REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
10-05-2012		

4. TITLE AND SUBTITLE

Reactive Swarm Formation Control Using Realistic Surface Vessel Dynamics and Environmental Effects

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Gerbino, Jacob Robert

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Naval Academy
Annapolis, MD 21402

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

Trident Scholar Report no. 404 (2012)

12. DISTRIBUTION/AVAILABILITY STATEMENT

This document has been approved for public release; its distribution is UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project presents a novel approach to Autonomous Surface Vessel coordination, applying adaptable swarm formation control to asset protection and escort missions. This approach includes realistic vessel dynamics and control inputs. Swarm coordination will be accomplished using a redundant manipulator formulation. The controller will allow for in-situ modification of the swarm formation based on varying environmental conditions.

15. SUBJECT TERMS

Autonomous Surface Vehicle, Redundant Manipulator Control, Realistic Dynamics, Swarm

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		79	19b. TELEPHONE NUMBER *(Include area code)*

U.S.N.A. --- Trident Scholar Project Report; no 404 (2012)

Reactive Swarm Formation Control Using Realistic Surface Vessel Dynamics and Environmental Effects

by

Midshipman 1/c Jacob R. Gerbino
United States Naval Academy
Annapolis, MD

_____
(signature)

Certification of Advisor Approval

Professor Bradley E. Bishop
Weapons and Systems Engineering Department

_____
(signature)

_____
(date)

Acceptance for the Trident Scholar Committee

Professor Carl E. Wick
Associate Director of Midshipman Research

_____
(signature)

_____
(date)

Abstract

This Trident Project focuses on the development of a controller for the coordination of a swarm of Autonomous Surface Vessels (ASVs) under mission and environmental constraints. For this research project, we first improved an existing Four Degree of Freedom (4DOF) vessel model. The model of a 360 metric ton patrol contains nonlinear hydrodynamics for the vessel's surge, sway, roll, and yaw motions. This model was modified to take into account environmental conditions including wind, waves and currents. Additionally, the control inputs of the model (propeller thrust, and desired ruder angle) were adapted for easier integration with a swarm level controller and additional nonholonomic motion constraints were applied to increase model fidelity. We then integrated this model into a simulated swarm of ASVs. It is the intention that this model will more accurately depict nonlinear vessel dynamics than did models used in previous ASV swarm studies.

Using a redundant robot manipulator formulation, the swarm controller enables ASVs to travel in a formation with the intent of protecting and escorting a hypothetical asset. To provide flexibility, the controller is capable of modifying its overall formation shape and mission parameters in response to varying environmental conditions. The Atlantic Center for the Innovative Design and Control of Small Ships, a research initiative sponsored by the Office of Naval Research, plans to apply techniques and methods developed in this study towards the construction and testing of a physical swarm of ASVs.

Keywords: Swarm, 4DOF, Redundant Manipulator, ASV

Table of Contents

Table of Figures

I. Introduction

Due to the extended wars in Iraq and Afghanistan, land and air based unmanned systems have become an important asset and a much addressed research area. The sea based equivalent of these assets, the Autonomous Surface Vessel (ASV), has lagged behind its counterparts and has yet to leap forward into wide use.

An ASV is a self-guided and unmanned watercraft that can be used to remotely carry out various missions. Interest in operational ASVs is quickly growing as the U.S. Navy looks for easier, cheaper and safer ways to perform dangerous or monotonous tasks.

The Office of Naval Research (ONR) has outlined five qualities of ASVs that make them appealing.[1] First is the decreased operating cost of unmanned systems. ASVs are more energy efficient than traditional marine assets, as they do not require systems that are essential to manned ships (climate control, interior lighting, ventilation, etc.). Thus an ASV can allocate a greater portion of its energy storage to mission-critical applications. Additionally, unmanned systems do not require basic utilities and services (plumbing, sewage, living spaces, medical spaces, etc.)

The second ASV quality of interest to ONR is enhanced coverage. Due to the increasing capabilities of electronic sensor systems, unmanned systems can maintain constant and up-to-date awareness of an environment. They can process a greater amount of information at a much faster rate than manual systems.

Third among the most appealing ASV qualities is productivity. The use of ASVs in routine or mundane missions means that manned platforms that would otherwise be carrying out those tasks can be free to carry out more complex or critical missions. In this case, ASVs can act as an effective force multiplier.

The fourth quality of interest for ASV use in the military is persistence. As stated earlier, ASVs are often more efficient than manned systems. This means that they can make their energy stores last longer, providing more on-station and on-task mission time. Additionally, ASVs can stay in operational areas longer than conventional surface ships because they do not require food and supply restocking (admitting the use of solar power for energy regeneration).

Finally, ASVs also provide decreased mission vulnerability. Using ASVs in dangerous missions keeps people and high-value assets out of harm's way. The expendability and relative low cost of ASVs vis-à-vis traditional marine assets means that they can take on high-risk missions, such as exploring hostile or dangerous environments to locate and disable explosives, or gathering intelligence in unfriendly waters. The loss of a drone at a cost of a few million dollars is much more acceptable than the loss of a manned vessel. ASVs will also be invaluable in the event of chemical, biological, or radiological (CBR) attack. Unmanned systems are relatively impervious to chemical and biological threats and can to some extent be shielded against radiation.

Despite their potential usefulness, ASVs have lagged behind their counterparts in the air, on land, and under the ocean's surface. This is primarily because of the complex difficulties that arise from operating at the interface between water and air. This is a very turbulent and difficult to model environment. The dynamics of systems traveling exclusively through the air or underwater are relatively well understood. Vessels traveling on the surface of the water, however, are exceptionally difficult to model. First, the hydrodynamics are complicated and

non-linear. Second, it is very difficult to simulate the effects of environmental disturbances such as wind and waves on a surface vessel.

In this study we developed a vessel maneuvering model for use in future swarm control studies. The model takes into account non-linear maneuvering of a vessel with a Four Degrees of Freedom (4DOF). In this case, the four degrees of freedom are surge, sway, roll and yaw. Once the model had been developed, we then developed a unit-level control system to control the non-holonomic ASV system using a control point method. We developed a swarm-level controller to coordinate a homogenous swarm of vessels. This novel control system allows for movement in formation and for modification of the swarm function based on environmental conditions.

In Section II, we discuss the Global and Body-Fixed coordinate frames used in this study. Section III covers the final 4DOF vessel model. This includes the control and environmental inputs as well as system outputs. Section IV covers the vessel-level controller used, which will allow it to follow commands given by the swarm-level controller. In Section V, we discuss methods used by the swarm-level controller. Section VI covers how environmental conditions affect the swarm. Section VII discusses methods of disturbance compensation. Section VIII discusses the results of our tests regarding these methods.

II. **Coordinate Frames**

Before going into an in-depth discussion of the work, it is first necessary to discuss the coordinate frames used in this study. Most of the visualizations of the vessel and swarm will consist of an overhead view. An example of this is provided in Figure 1. In this figure, notice that North is aligned towards the top of the page. Please also note that the axis of the plot oriented towards the top of the page is labeled “X” while the axis oriented to the right of the page is labeled “Y”. This is contrary to how most people conceive of a coordinate plane.

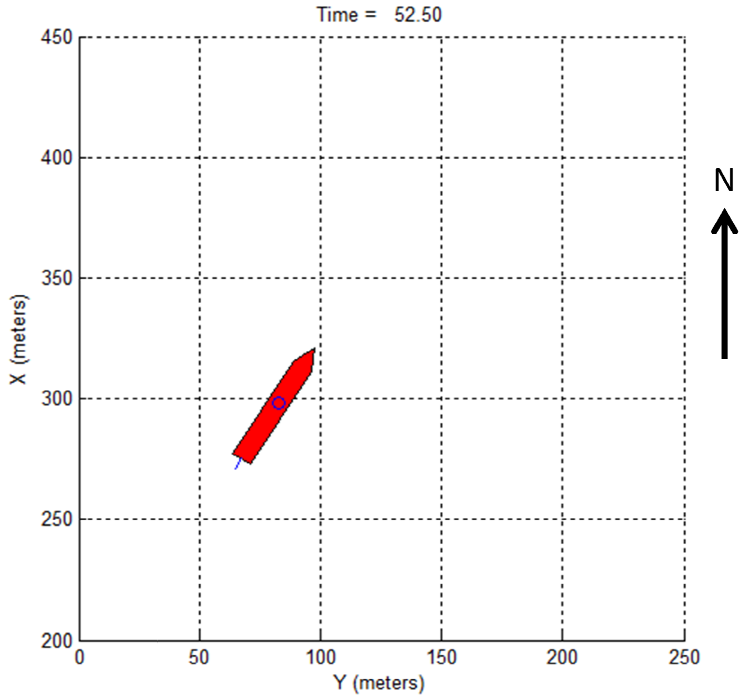


Figure 1: Overhead View of Vessel

The coordinate plane used for most applications has the two axes reversed, as is shown in Figure 2(a). This is the typical Cartesian plane. Since this is a two dimensional projection, the z-axis is not generally recognized but can be imagined as an axis perpendicular to the surface of the page with the positive axis extending out of the page towards the reader. This is more than adequate for most situations, however, in applications concerning navigation, orientation of the X, Y, and Z axis must be changed.

When dealing with angles in the X-Y plane and about the X-axis, the angle is measured in relation to the positive X-axis. Due to the right-hand rule, the angles increase in a counterclockwise direction. If applied to navigation, a direction of 000° True heading (heading angle given in degrees is generally expressed as a three digit number) would be pointed towards the East and the heading angle would increase as the observer turned counterclockwise. This is contrary to how our navigational system works. A direction of 000° True heading points to the North and the heading angle increases as the observer turns counterclockwise.

To reconcile the coordinate system with the generally accepted navigational system, this model uses a North-East-Down(NED) reference frame. In Figure 2(b), this coordinate frame has been illustrated. Here, North and South directions are represented by the positive X and Y axis, respectively. Consequently, the positive Z-axis must be pointed into the page, away from the reader. This creates a Global fixed frame of reference which resembles our current navigational conventions.

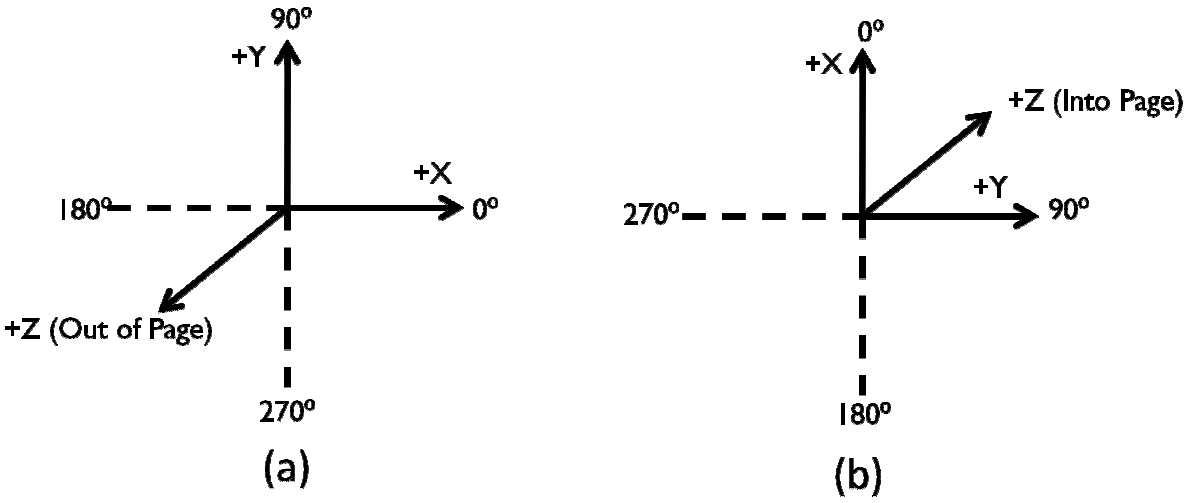


Figure 2: (a) Conventional Coordinate Frame versus (b) North-East-Down Coordinate Frame

For the body-fixed frame, we used a similar coordinate frame. The vessel's vertical axis (Heave) was oriented downward. This allowed relative bearing for the vessel to be expressed in the same fashion as an actual vessel, with 0° relative to the ship being located directly in front of the vessel's bow (in the direction of the positive X axis), and increasing in the clockwise direction. Rotation about the heave axis is called yaw. Extending directly ahead of the vessel is the surge axis. Rotation about this axis is called roll. Extending directly to the right of the vessel is the sway axis. Rotation about this axis is called pitch. These terms are illustrated in Figure 3.[2]

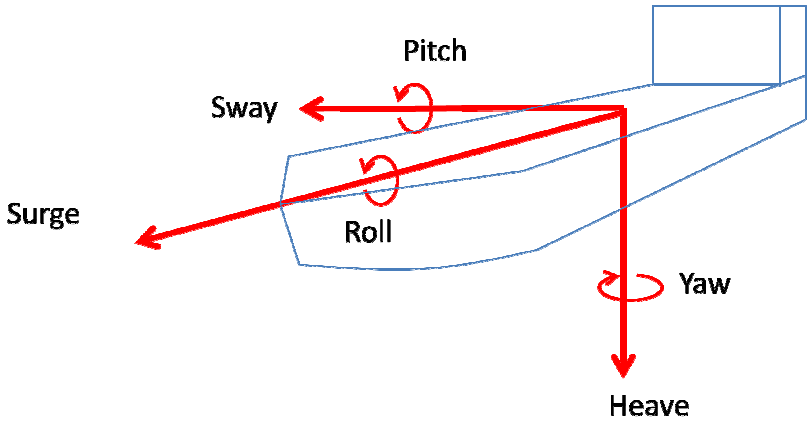


Figure 3: Body-Fixed Coordinate Frame

In this case, it is also important to define how the yaw angle, or heading, of the vessel is measured. The yaw angle, expressed as ψ, is defined as the angle between the positive x axis of the Global Frame and the surge axis of the vessel.

III. Model

This project began by evaluating existing maneuvering modes and investigation how they could fit into the swarm study. We required a non-linear vessel model that took into account at least four degrees of freedom (surge, sway, roll, and yaw are the degrees of key interest).

In order to create a descriptive representation of a vessel's handling characteristics, dynamics-based representations are often used. Researchers today often use some variation of the following equation to model the motion of the *i*th vessel in a group while taking into account environmental and control dynamics:

$$M_i\dot{v}_i = -\boldsymbol{C}(v_i)v_i - D_iv_i + \tau_i + J(\psi_i)^T\tau_{ci}$$

where M_i is the inertia matrix

$$M_i = \begin{bmatrix} m_{11i} & 0 & 0 \\ 0 & m_{22i} & m_{23i} \\ 0 & m_{32i} & m_{33i} \end{bmatrix}$$

D_i is the damping matrix

$$D_i = \begin{bmatrix} d_{11i} & 0 & 0 \\ 0 & d_{22i} & d_{23i} \\ 0 & d_{32i} & d_{33i} \end{bmatrix}$$

C_i is the combined Coriolis and Centrifugal matrix

$$C_i(v_i) = \begin{bmatrix} 0 & 0 & -m_{22}v_i - m_{23}r \\ 0 & 0 & m_{11}u_i \\ m_{22}v_i + m_{23}r & -m_{11}u_i & 0 \end{bmatrix}$$

$J(\psi_i)$ is the system's Jacobian matrix

$$\boldsymbol{J}(\psi_j) = \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and τ_{Ci} encodes environment effects

$$\tau_{Ci} = [\tau_{cu} \quad \tau_{cv} \quad \tau_r].$$

Finally, τ_i is the force in the direction of surge and moment in the direction on yaw provided by the vessel's engines.[3][4]

$$\tau_i = [\tau_{ui} \quad 0 \quad \tau_{ri}]$$

The τ_i term represents a major shortcoming of current models. The model assumes that the vessel's actuators can independently provide a force to control the vessel's surge force and yaw torque. This is not the case with most real vessel designs, and certainly not with existing ASVs. On most boats, a vessel's turn is dependent on a transverse force developed by the rudder. The magnitude of the transverse force on a rudder is a function of water speed flowing over the rudder and the rudder angle relative to the boat's center line.[5] The practical implications of this are that a boat cannot turn while stationary and a boat's turn radius varies with vehicle speed and rudder angle. The turn radius of a boat is illustrated in Figure 4. Advance is the distance traveled during the turn in the direction of the original heading of the boat and transfer is the distance traveled during the turn perpendicular to the original heading of the boat.[6]

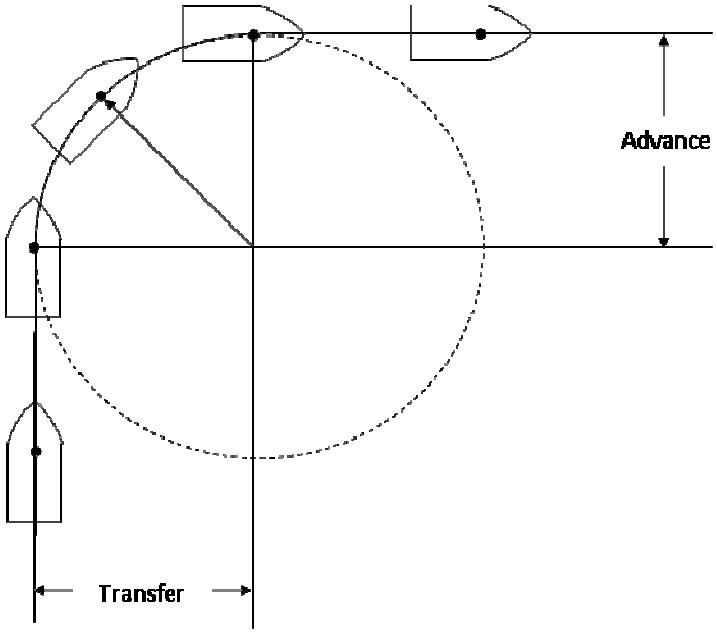


Figure 4: Turn Circle of a Vessel

A slightly different system model was developed to take into account the turning dynamic in boats.[7] The term τ_i is replaced with $\boldsymbol{B_i f_i}$ where

$$B_i = \begin{bmatrix} b_{11i} & 0 \\ 0 & b_{22i} \\ 0 & b_{32i} \end{bmatrix}$$

is the actuator matrix, which takes into account the effect of the boat's rudder on sway and $f_i = [T_{ui}, T_{ri}]^T$ is the control input vector. The term T_u is the thrust developed by the boat's propulsion, and T_r is the rudder deflection. This model, while better than the nominal, still allows unrealistic maneuvering.

In addition to being linearized for the sake of simplicity, models such as these fail to take into account more than surge, sway and yaw of the vessel. In these cases, the roll of the vessel is something we are interested in.

Our study required a nonlinear maneuvering model, which allowed for realistic control inputs, rudder angle and propeller thrust. This model also had to integrate the effects of the nautical environment; in this case, current, wind and waves. Shown in Figure 5 is a depiction of the vessel maneuvering system we developed for the study. It contains the vessel's hydrodynamics as well as the rudder and thrust machinery and the current and wind effects on the vessel.

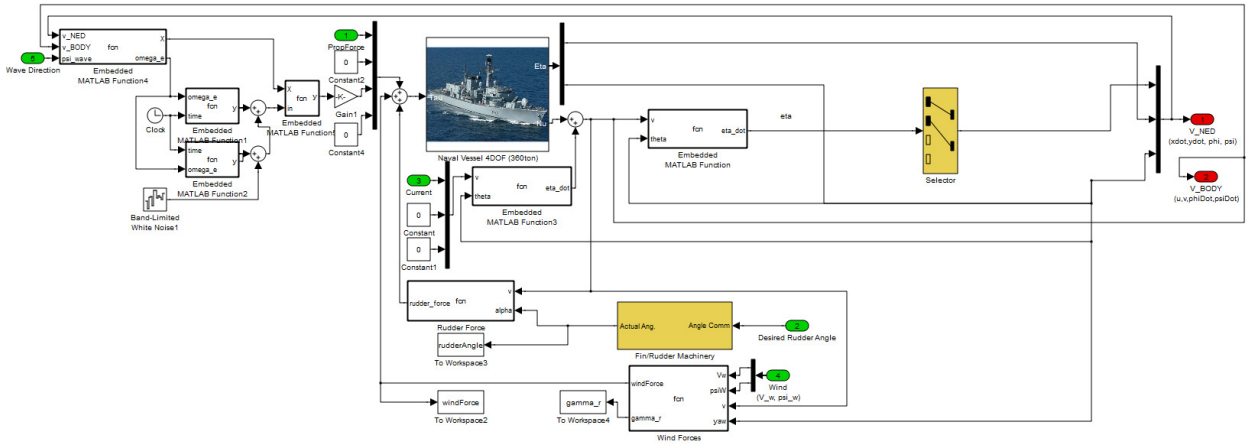


Figure 5: Vessel dynamics and control model in Simulink

The vessel maneuvering system was developed to be contained in a Simulink® subsystem block. This allows for easy integration with control systems. Additionally, it allows for easy duplication for simulations when more than one of these vessels will be used. The mask developed for the simulated subsystem block is shown in Figure 6. The inputs and outputs of the system will be discussed later in this section.

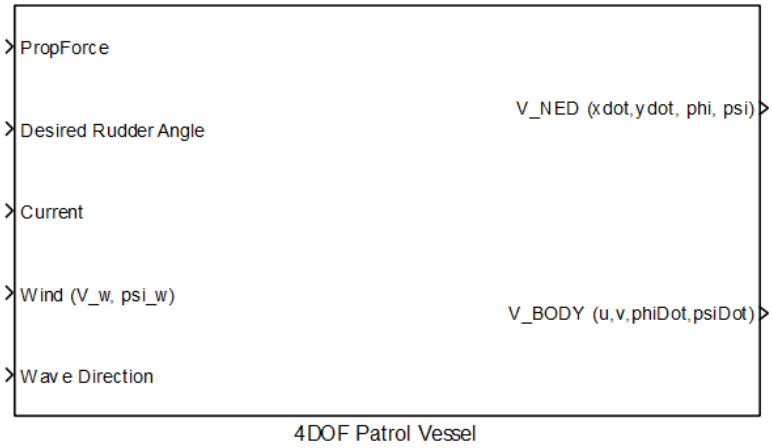


Figure 6: Simulink Block in which model dynamics are contained

a. Maneuvering Dynamics

This model is a 4DOF representation of a 360 metric ton patrol vessel developed by researchers Blanke and Perez for the Marine Systems Simulator.[8] The dynamics of the vessel are contained in the "Naval Vessel 4DOF (360ton)" block, shown in Figure 7.

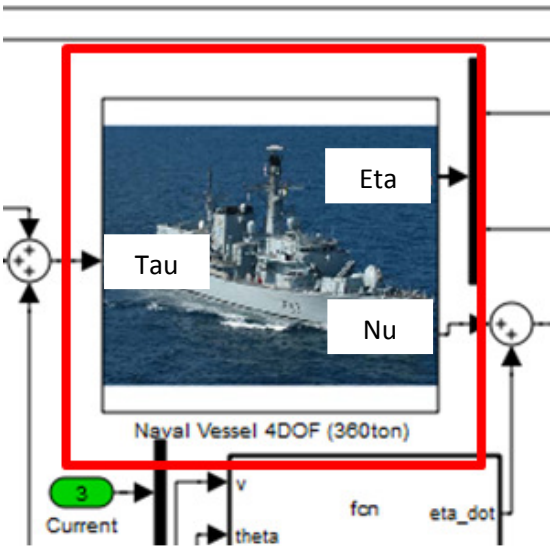


Figure 7: "Naval Vessel 4DOF (360ton)" Simulink Block Containing Vessel Dynamics

The input to the block "Tau" (left middle), τ, represents the forces acting on the model due to wind, the rudder, and the propeller

$$\tau = \begin{bmatrix} X \\ Y \\ K \\ N \end{bmatrix}$$

Where:

- X is the sum of the forces in the vessel's surge direction
- Y is the sum of the forces in the vessel's sway direction
- K is the moments of the forces about the vessel's surge axis(Roll Moment)
- N is the moments of the forces about the vessel's heave axis(Yaw Moment)

The "Naval Vessel 4DOF (360ton)" block has two outputs. The first being "Eta" (right top), $\eta = [\phi \quad \psi]'$, where ϕ and ψ represent the vessel's roll and yaw angles, respectively. The second output is the vessel's velocity in the body frame "Nu" (right bottom), ν^B, defined as:

$$\nu^B = \begin{bmatrix} u \\ v \\ p \\ r \end{bmatrix}$$

Where:

- u is vessel's velocity in the surge direction
- v is vessel's velocity in the sway direction
- p is the angular velocity about the vessel's surge axis(Roll Moment)
- r is the angular velocity about the vessel's heave axis(Yaw Moment)

For use in the study, v^B must be converted into a velocity with respect to the fixed Global frame:

$$v^B = \begin{bmatrix} \dot{x} \\ \dot{y} \\ p \\ r \end{bmatrix}$$

Where $\dot{x}$ is vessel's velocity along the Global x-axis and $\dot{y}$ is the sum of the forces in the Global y-axis. This rotation is accomplished using the rotation matrix, R_B^G to produce v^G:

$$R_B^G = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$v^G = R_B^G v_B$$

Where ψ is the vessel's heading, defined as the angle between the Global x-axis and the vessel's surge axis.

b. Thrust

For the thrust system, shown in Figure 8, we limited the forward thrust that the vessel could provide to about 1.7e5 N. This is the thrust required to propel the vessel to just above *hull speed* (about 9.0 m/s). Beyond hull speed the vessel's hull will go into a planing mode. In this regime, the dynamic model being used is not valid, as it was developed for a vessel in displacement mode. We have also imposed limitations on thrust to prevent the vessel from being propelled in reverse by its engines. The model was not developed to represent motion astern. On open water, vessels will rarely go in reverse, instead opting to turn using the rudder when an extreme change in course is needed. Changes in desired thrust are modeled as a first order transfer function with a time constant of 1s and a settling time of 5s. This allows us to take into account delays caused by the engine and propeller machinery.

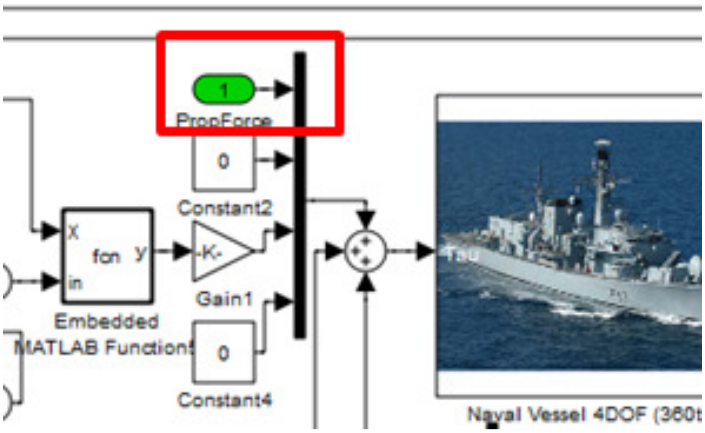


Figure 8: Thrust Input

c. Rudder

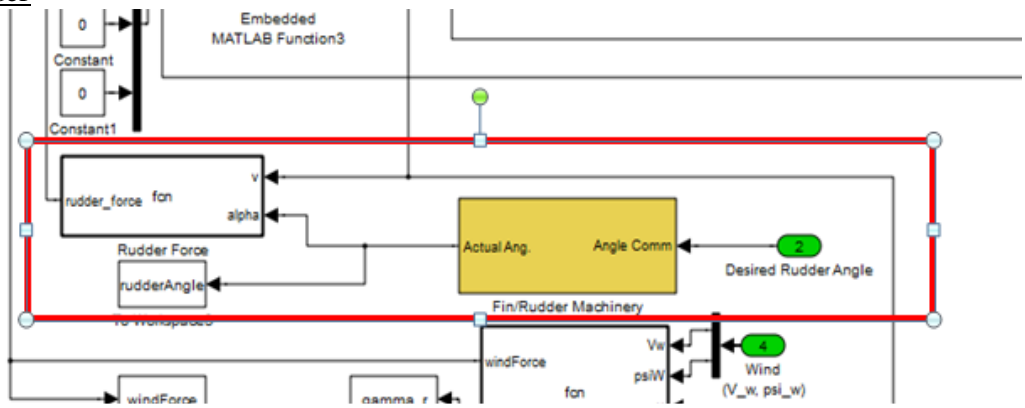


Figure 9: Rudder System

To include an accurate representation of a rudder (Figure 9), we were able to adapt a previously existing model [8] to suit our needs. This model calculates forces and moments developed from the rudder as a function of rudder angle and vessel speed. Also accounted for in this model is the rudder machinery. Constraints were placed on the rudder's ability to react to a change in rudder command. The transient response of the rudder to a new desired rudder angle has a time constant of 0.2s and a maximum rate of 20 degrees per second. [8]

The rudder had a stall angle of 23 degrees. The rudder's lift coefficient was modeled to increase linearly until it reached 23 degrees. Beyond that point, the rudder can displace further, however, no increased is obtained. This relationship is displayed below:

$$C_r(\delta_r) = \begin{cases} -23^\circ * dC_r, & -45^\circ \leq \delta_r < -23^\circ \\ \delta_r * dC_r, & -23^\circ \leq \delta_r \geq 23^\circ \\ 23^\circ * dC_r, & 23^\circ < \delta_r \leq 45^\circ \end{cases}$$

In Figure 10, we have plotted the value of the rudder's lift coefficient versus the angle of the rudder with respect to the ship's center-line.

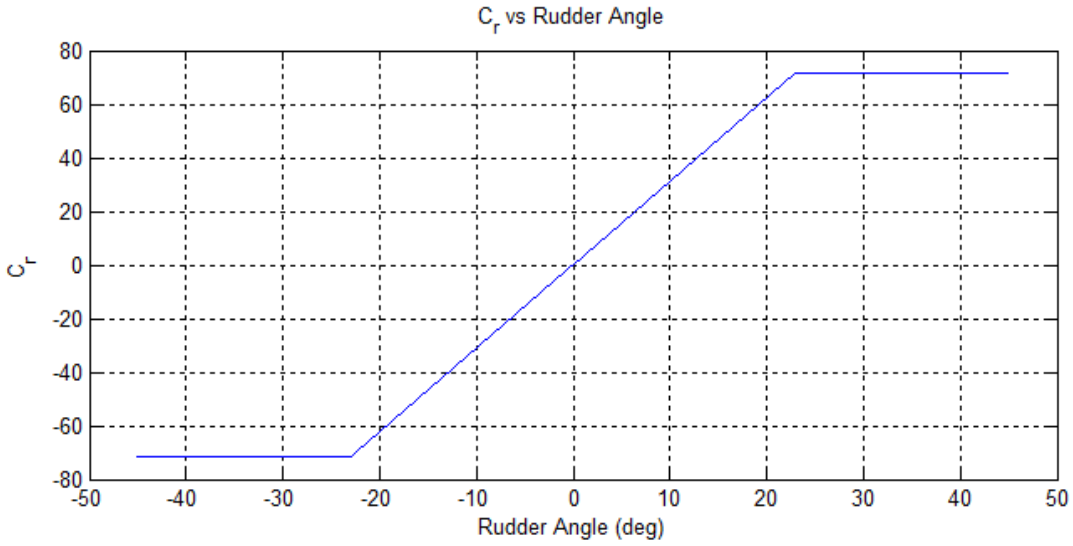


Figure 10: Graph of C_r (δ_r) versus rudder angle (δ_r)

The rudder model takes into account the forces and moments produced in relation to all four of the vessel's degrees of freedom. These forces and moments are calculated according to the following:

$$\tau_{rudder} = \begin{bmatrix} X_{rudder} \\ Y_{rudder} \\ K_{rudder} \\ N_{rudder} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\left(\frac{C_r(\delta_r)^2}{0.9\pi C_{\text{aspect}}} + C_{D0}\right)\rho_{water} v^2 A_{rudder} \\ \frac{1}{2} C_r(\delta_r)\rho_{water} v^2 A_{rudder} \\ -\frac{1}{2} C_r(\delta_r)\rho_{water} v^2 A_{rudder}\left(V_{CG} - \frac{1}{2}S\right) \\ -\frac{1}{2} C_r(\gamma_r)\rho_{water} A_{rudder} L_{CG} \end{bmatrix}$$

Where:

- C_{aspect} is the aspect ratio of the rudder (3)
- C_{D0} is the drag coefficient of the rudder when $\delta_r = 0$ (0.0065)
- ρ_{water} is the density of sea water ($1025 \frac{kg}{m^3}$)
- A_{rudder} is the area of the rudder ($1.5m^2$)
- V_{CG} is the height of the vessel's center of gravity above the rudder's geometric center (3.36 m)
- S is the span of the rudder (1.5m)
- L_{CG} is the distance from the rudder's geometric center to the vessel's center of gravity along the vessel's surge axis (19.82m) [8]

d. Current

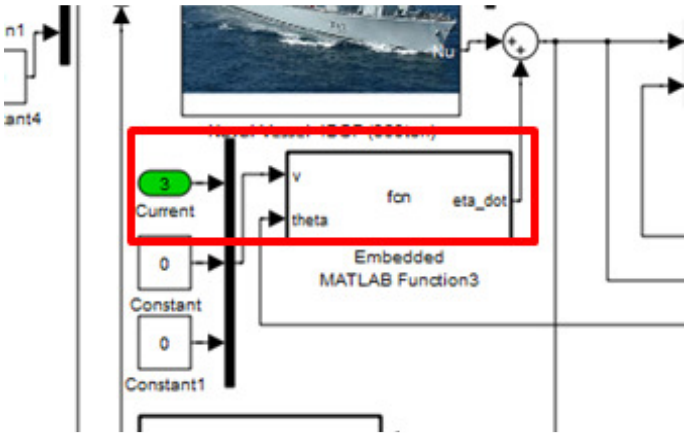


Figure 11: Current Input

Current was represented as additive to the vessel's speed. Since the vessel maneuvers in relation to the ocean, this is a reasonable approximation when using constant or slowly varying currents. [9] The velocity of the current in relation to the Global frame is expressed as $v^G_{Current} = [v_{xCurrent} \quad v_{yCurrent}]^T$ where $v_{xCurrent}$ is the X-component of the current velocity and $v_{yCurrent}$ is the Y-component of the velocity. These velocities must be expressed in the body-fixed frame in order to be added to the vessel's velocity. This is done according to the following equations:

$$v^B_{current} = R^B_G \begin{bmatrix} v^G_{Current} \\ 0 \\ 0 \end{bmatrix}$$

$$v^B_{Vessel} = v^B + v^B_{current}$$

$$v^G_{Vessel} = R^G_B v^B_{Vessel}$$

Where:

- $v^B_{current}$ is the velocity of the current in relation to the body-fixed frame
- R^B_G is the Global-to-Body rotation matrix and is defined as $R^B_G = (R^G_B)^T$
- v^B is an output of the "Naval Vessel 4DOF (360ton)" block as defined earlier
- v^B_{Vessel} is the total velocity of the vessel expressed in the body-fixed frame
- v^G_{Vessel} is the total velocity of the vessel expressed in the Global frame

e. Wind

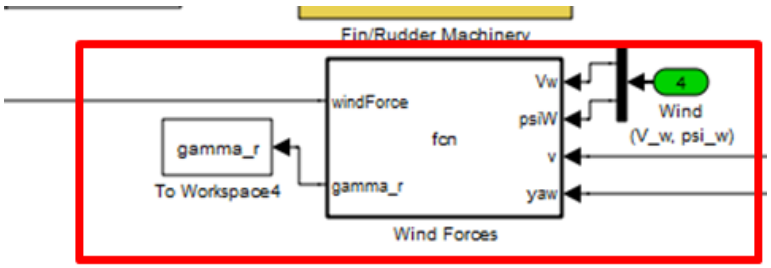


Figure 12: Wind Input

This model takes into account the effects of wind on the vessel's motion. For accuracy, it is necessary to take into account the vessel's own motion in relation to the wind and, from those factors, calculate the apparent wind. For example, if a vessel was moving in a forward direction at 5m/s in still air, an observer on the vessel would measure a 5m/s wind moving across the vessel from bow to stern (or from 000℟). Similarly, the force produced by the drag on the vessel would be comparable to that produced by a 5m/s wind moving across the vessel when it is at rest.

The apparent wind velocity of the vessel is dependent on the wind's speed and direction relative to the vessel's velocity and yaw angle. The apparent wind speed and direction relative to the vessel's frame of reference can be calculated using the equations below: [2]

$$\gamma = \psi_{wind} - \psi$$
$$u_R = V_{wind}\cos(\gamma) + u$$
$$v_R = V_{wind}\sin(\gamma) + v$$
$$\gamma_R = \tan^{-1}\frac{v_R}{u_R}$$
$$V_R = \sqrt{u_R^2 + v_R^2}$$

Where:

- γ is the wind angle relative to the body-fixed frame
- ψ_{wind} is the direction of the wind in the Global frame
- ψ heading (or yaw angle) of the vessel
- V_{wind} is the speed of the wind
- u_R is the apparent velocity of the wind across the vessel's surge axis
- v_R is the apparent velocity of the wind across the vessel's sway axis
- u is the surge speed of the vessel
- v is the sway speed of the vessel
- γ_R is the apparent wind angle relative the vessel's heading

The model for calculating wind's effect on surge, sway, and yaw was developed using the method of Isherwood (1972) where: [10]

$$\tau_{wind} = \begin{bmatrix} X_{wind} \\ Y_{wind} \\ K_{wind} \\ N_{wind} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}C_x(\gamma_r)\rho_{air}V_r^2 A_T \\ \frac{1}{2}C_y(\gamma_r)\rho_{air}V_r^2 A_L \\ 0 \\ \frac{1}{2}C_n(\gamma_r)\rho_{air}A_T L \end{bmatrix}$$

$$C_X(\gamma_r) = -\left(A_0 + A_1 2\frac{A_L}{L^2} + A_2 2\frac{A_T}{B^2} + A_3\frac{L}{B} + A_4\frac{S}{L} + A_5\frac{C}{L} + A_6 M\right)$$

$$C_Y(\gamma_r) = B_0 + B_1 2\frac{A_L}{L^2} + B_2 2\frac{A_T}{B^2} + B_3\frac{L}{B} + B_4\frac{S}{L} + B_5\frac{C}{L} + B_6\frac{A_{SS}}{A_L}$$

$$C_Z(\gamma_r) = C_0 + C_1 2\frac{A_L}{L^2} + C_2 2\frac{A_T}{B^2} + C_3\frac{L}{B} + C_4\frac{S}{L} + C_5\frac{C}{L}$$

Where:

- $C_x(\gamma_r)$ is the drag coefficient for the drag force along the vessel's surge axis
- $C_y(\gamma_r)$ is the drag coefficient for the drag force along the vessel's sway axis
- $C_n(\gamma_r)$ is the drag coefficient for the wind-induced yaw moment
- ρ_{air} is the density of air ($1.225\frac{kg}{m^3}$)
- V_r is the apparent wind velocity
- A_T is the transverse area of the vessel
- A_L is the lateral area of the vessel
- L is the length of the vessel
- S is the length of the parameter of the lateral projection excluding the waterline
- A_{SS} is the area of the superstructure
- C is the distance from the bow of the lateral projected area
- M is the number of masts separate from the superstructure
- The values of A_i, B_i, and C_i are must be interpolated from a table of Isherwood Values (Appendix A)

In order to apply Isherwood's method to the current vessel model, a hypothetical profile was created that allows us to provide values for $C_x(\gamma_r)$, $C_Y(\gamma_r)$, $C_n(\gamma_r)$. Figures 13 and 14 show the lateral and transverse views, respectively, of the hypothetical vessel profile. These simplified profiles were used to calculate the drag coefficients of the vessel.

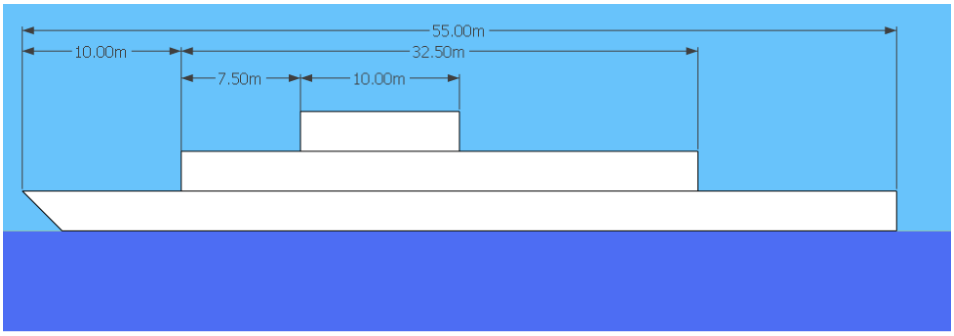


Figure 13: Lateral Projection of Vessel Profile

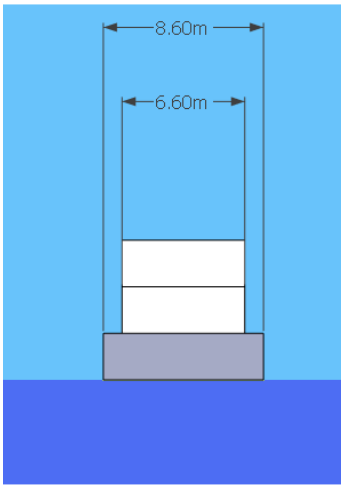


Figure 14: Transverse Projection of Vessel Profile

It is important to note that the effects of wind on a vessel change as the apparent wind angle changes. The Isherwood model was specifically chosen because it takes into account variations of the vessel's drag coefficients with respect to relative wind angle. For our vessel the coefficients were measured at various angles and the results are shown in Figure 15.

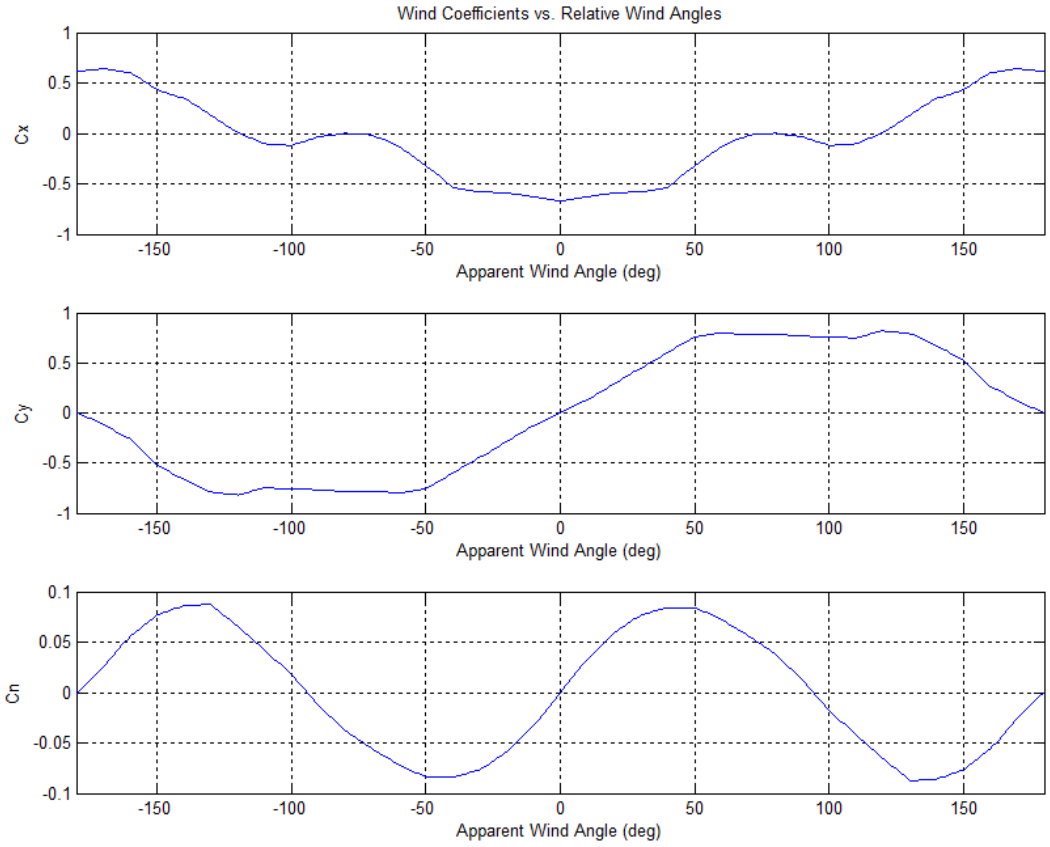


Figure 15: Wind Drag Coefficients versus Apparent Wind Angle

f. Waves

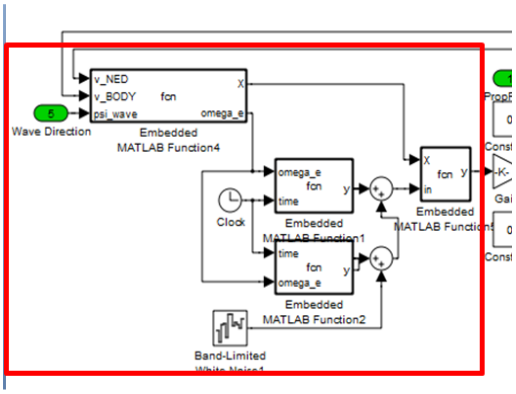


Figure 16: Wave System

To take into account the effect of surface waves on the vessel, we adopted a previously existing model produced by Benstead in order to simulate the roll moment on vessels. In that model, the time-varying height of a wave function was expressed as the addition of two sinusoids and a white noise function. This model did not take into account the effects of vessel speed or orientation. [11]

$$\chi = \psi_{wave} - (\psi - \pi)$$

$$\omega_e = \omega - \frac{\omega^2 u}{g}\cos(\chi)$$

$$h_{wave} = \mathrm{A}_1 \sin(\omega_e t) + \mathrm{A}_2 \sin\left(\frac{1}{2}\omega_e t + \phi\right) + G$$

$$K_{wave} = k_{wave} * h_{wave} \sin(\chi)$$

Where:

- χ angle between the vessel's negative surge axis and the direction of wave propagation
- ψ_{wave} is the direction of wave front propagation
- ψ is the heading of the vessel
- ω_e is the encounter frequency
- ω is the peak frequency of the wave (.6970 $\frac{rad}{s}$)
- u is the surge speed of the vessel
- g is acceleration due to gravity (9.81 $\frac{m}{s^2}$)
- h_{wave} is the wave height
- A_1 is the amplitude of peak frequency signal (1.625)
- t is time
- A_2 is the amplitude of the one half peak frequency signal (1.00)
- ϕ is the phase offset of the one half peak frequency signal ($\frac{3\pi}{2} rad$)
- G is a Gaussian white noise function
- K_{wave} is the wave-induced roll moment

- k_{wave} is the constant relating wave height to wave-induced roll moment ($0.5e5\frac{Nm}{m}$)

It is important to note that encounter frequency varies depending on the orientation and speed of the vessel with relation to the wave front. Due to the Doppler Effect, the perceived frequency of a wave increases when the vessel is moving opposite the direction of wave propagation and decreases when the vessel is moving with the direction of propagation. [12]

The magnitude of the effect of waves on vessel's roll varies with the vessel's orientation relative to the direction of wave propagation. When the waves are hitting the vessel directly head-on or from the rear, there should be almost no roll moment produced. When waves are hitting the vessel directly from the sides, however, the maximum amount of roll moment is produced.

g. Model Validation

i. Turn Dynamics

In our first test, we looked at the turn dynamics of a ship in the absence of wind or currents. We set the vessel to travel in a straight direction and then after the speed of the vessel had reached steady-state, commanded the rudder to right-full(-45°) this caused the simulated vessel to turn to the right. Figure 17 depicts the vessel and its turn circle.

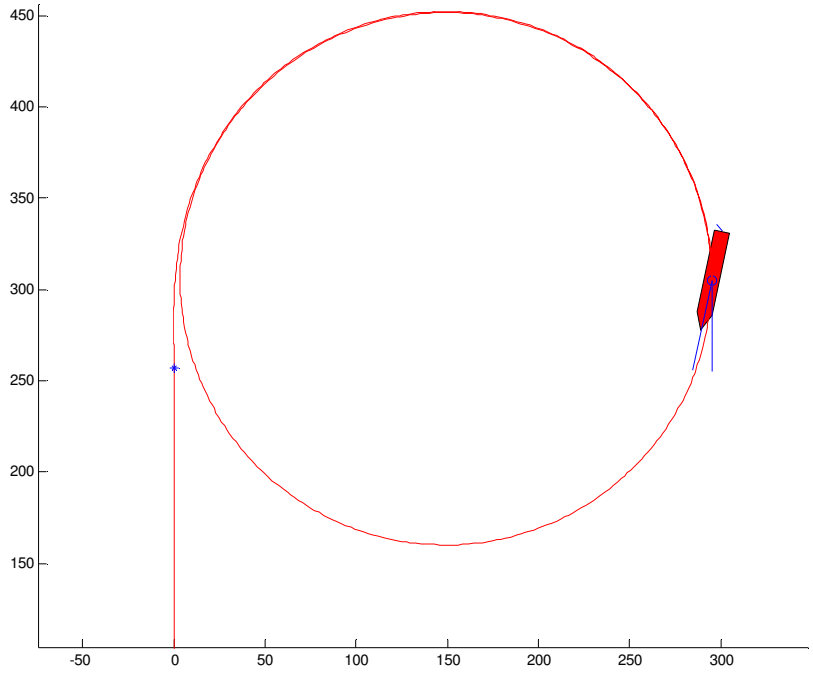


Figure 17: Turn Circle of Vessel

Figure 18 is a closer view of the ship in Figure 17. Here drifting of the vessel cans be seen. The heading of the vessel is not tangent to the turn circle;

rather, the vessel is turned into the circle. Such would be the case of a real vessel in a turn.

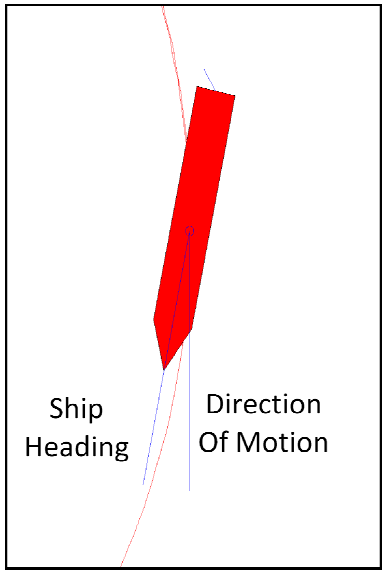


Figure 18: Close-Up View of Vessel in Turn

Figure 19 is a closer view of the path taken by the vessel immediately following a rudder command. A blue asterisk has been plotted to show the location of the vessel's center of gravity at the moment the vessel's rudder was turned. As seen below, the vessel's center of gravity actually moves slightly to the vessel's port before turning to starboard. Though counter-intuitive, this is characteristic of a real vessel's turn circle. This is because the rudder generates a force in the sway direction as well as a yaw moment. [2] Additionally, the rudder moment did not instantly impose a yaw rate on the vessel. The vessel continued to move forward before turning. This also is consistent with the maneuvering of surface vessels.

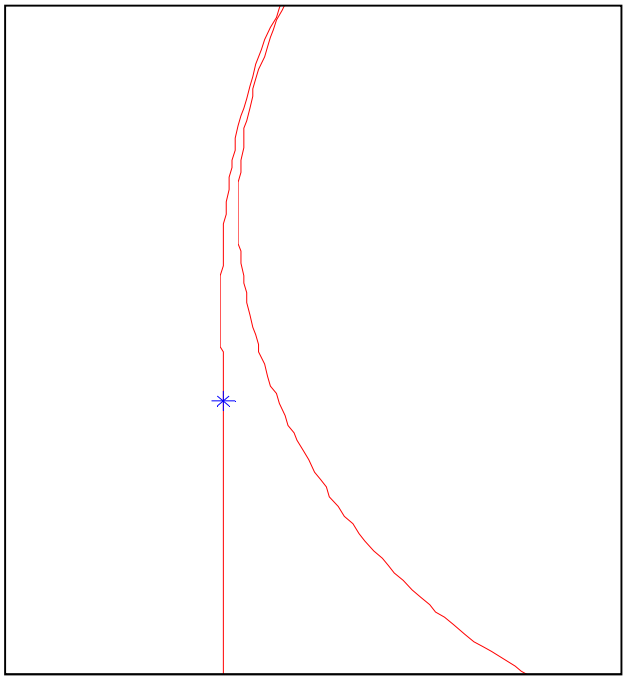


Figure 19: Beginning of Vessel Turn

ii. Wind Effects

To verify that our wind force equations were accurately integrated into the system model, we tested the vessel in a variety of situations with varying wind speeds.

1. Trial 1

To test the effect of wind on the maneuvering of the vessel, we used a simple proportional controller to direct the rudder with the purpose of maintaining the vessel's heading. The control law for the vessel's yaw is defined as:

$$\delta_r = k_r(\psi_d - \psi)$$

Where the $k_r = -15$, ψ_d is the vessel's desired heading and ψ is the vessel's actual heading. As can be seen in Figure 20, the vessel's course is moved in the direction of the wind with the magnitude of this effect becoming greater as the speed of the wind was increased.

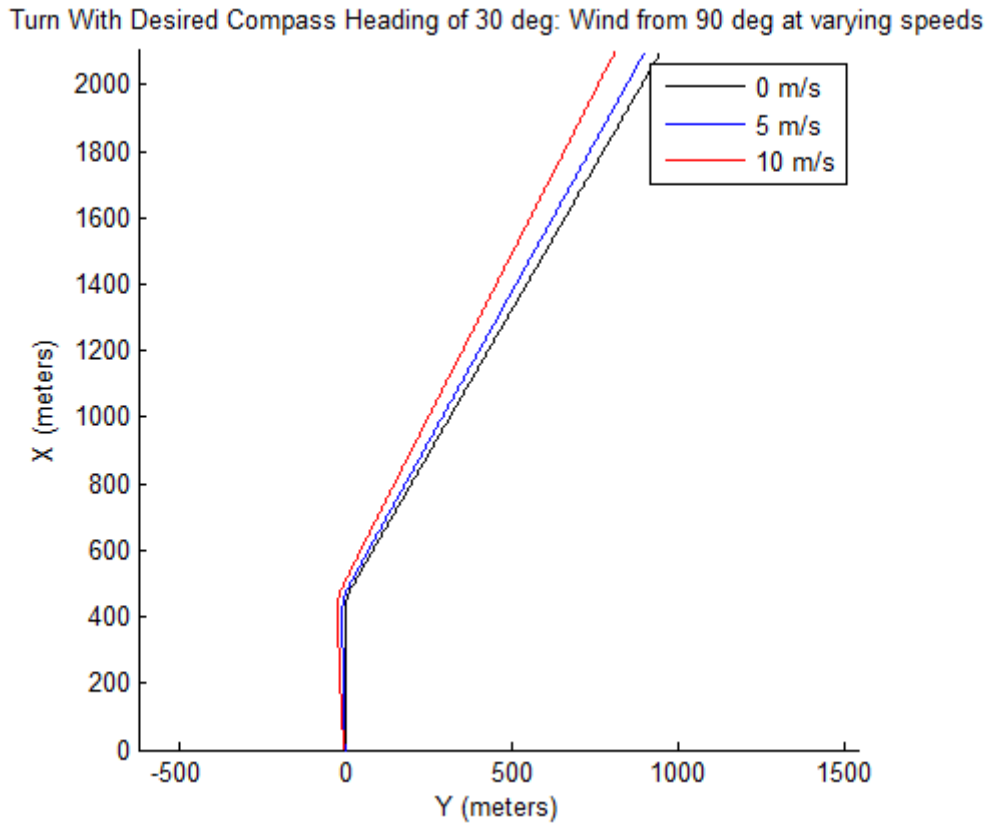


Figure 20: Vessel Movement at Various Wind Speeds

2. Trial 2

To further test the wind model, we simulated a full 360 turn of the vessel under varying wind speeds and then plotted the path of the vessel. These plots can be seen in Figure 21. The direction of the wind in these tests was from 090°, or blowing towards west. We found that as the wind speed increases, it pushed the turn circle of the craft further towards the left. This is consistent with the behavior of a true turn circle.

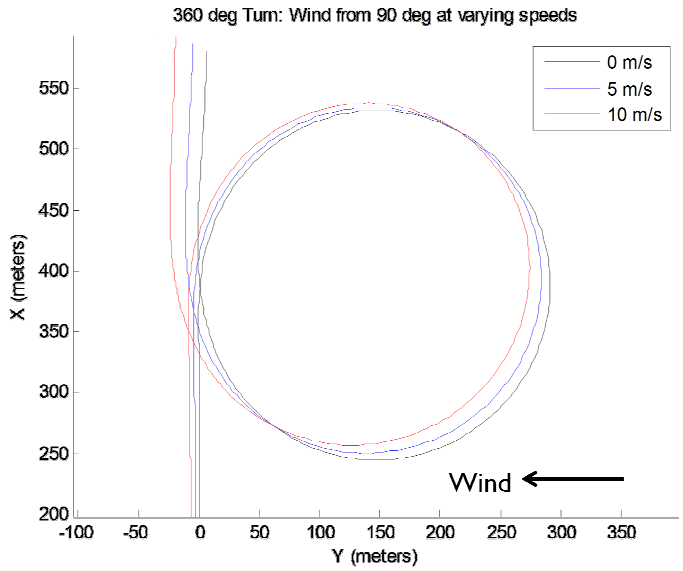


Figure 21: Vessel Turn Circle under Varying Wind Conditions

iii. Wave Effects

To verify the frequency shift that occurs as result of the Doppler Effect, we ran a simulation in which waves were coming from an angle of 045° True and the vessel was facing due North. We compared the roll angles of the vessel while the vessel attempted to travel north at 2m/s versus when it was at rest. The results were plotted in Figure 22. As can be seen, vessel roll angles are almost exactly the same at the beginning of the simulation. As the moving vessel's speed increased, however, roll oscillations began to lead the roll oscillations of the vessel at rest. This means that the encounter frequency of the waves is increasing as expected.

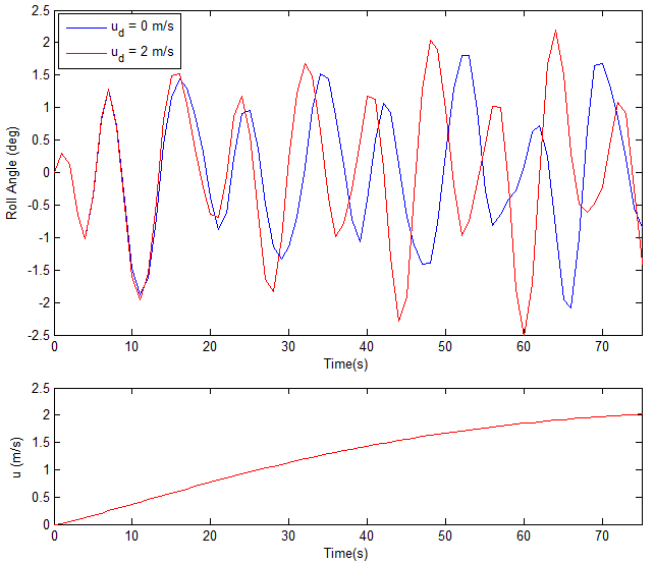


Figure 22: Roll Angle and Vessel Speed vs Time

We also need to test the opposite case. For this simulation, waves were coming from an angle of 135° True and the vessel was again facing north. In this case, we also compared roll angles of the vessel while the vessel attempted to travel north at 2m/s versus when it was at rest. These results are shown in Figure 23. As can be seen, the vessel roll angles are almost exactly the same as at the beginning of the simulation. As the moving vessel's speed increased, however, roll oscillations began to lag behind the roll oscillations of the vessel at rest. This means that the encounter frequency of the waves is decreasing as expected.

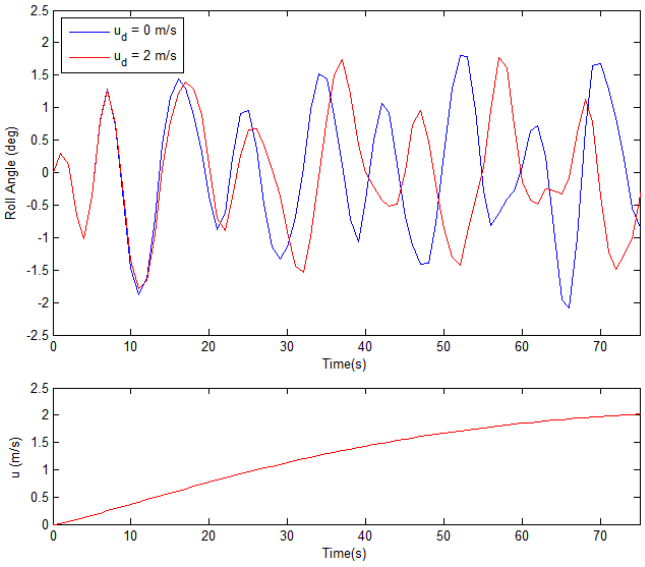


Figure 23: Roll Angle and Vessel Speed vs Time

IV. Vessel-Level Control

To control the trajectory of the vessel we will designate a control point on the vessel which is along the vessel's center line and not on the vessel's center of gravity. This point is given a desired velocity in the Global Frame ($v_d^G = [v_{dx}, v_{dy}]^T$). This is then reduced into component vectors in the body frame along its surge and sway directions ($v_d^b = [v_{di}, v_{dj}]^T$) using the following equation:

$$v_d^b = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} v_d^G$$

This process is shown graphically in Figure 24. [13]

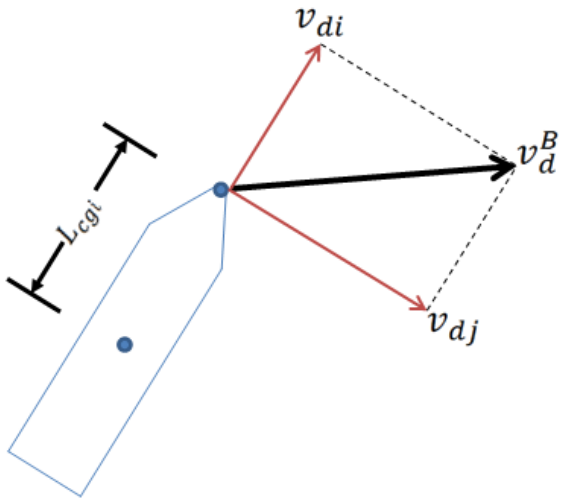


Figure 24: Resolution of v_d^b into v_{di} and v_{dj}

The value v_{di} will be the vessel's desired surge velocity. Thrust, X_{Thrust}, is provided according to the following control law:

$$X_{Thrust} = k_{sp}(v_{di} - v_i) + k_{si}\int (v_{di} - v_i)\, dt$$

The integral term in the above equation ensures that the steady-state error between the desired velocity and the actual velocity gradually declines to zero. The term, v_{dj}, then is used to calculate the desired yaw rate, r_d, where $r_d = v_{dj}/L_{cgi}$, according to the following control law:

$$\delta_r = k_r(r_d - r)$$

In this case, δ_r represents the desired rudder angle, which will then be fed into the vessel system block.

To demonstrate the single-vessel control system, we created a simulation in which the vessel was given desired value in the Global frame ($v_d^G = [2,5]^T \frac{m}{s}$) The vessel started from rest with its center of gravity located at (0,0) in the Global frame. Figure 25 contains images from the simulation at various times. As can be seen from Figure 25, the vessel begins to move forward, and as the vessel accelerates, the vessel begins to turn. The path of the vessel settles out when it is pointed in the correct direction.

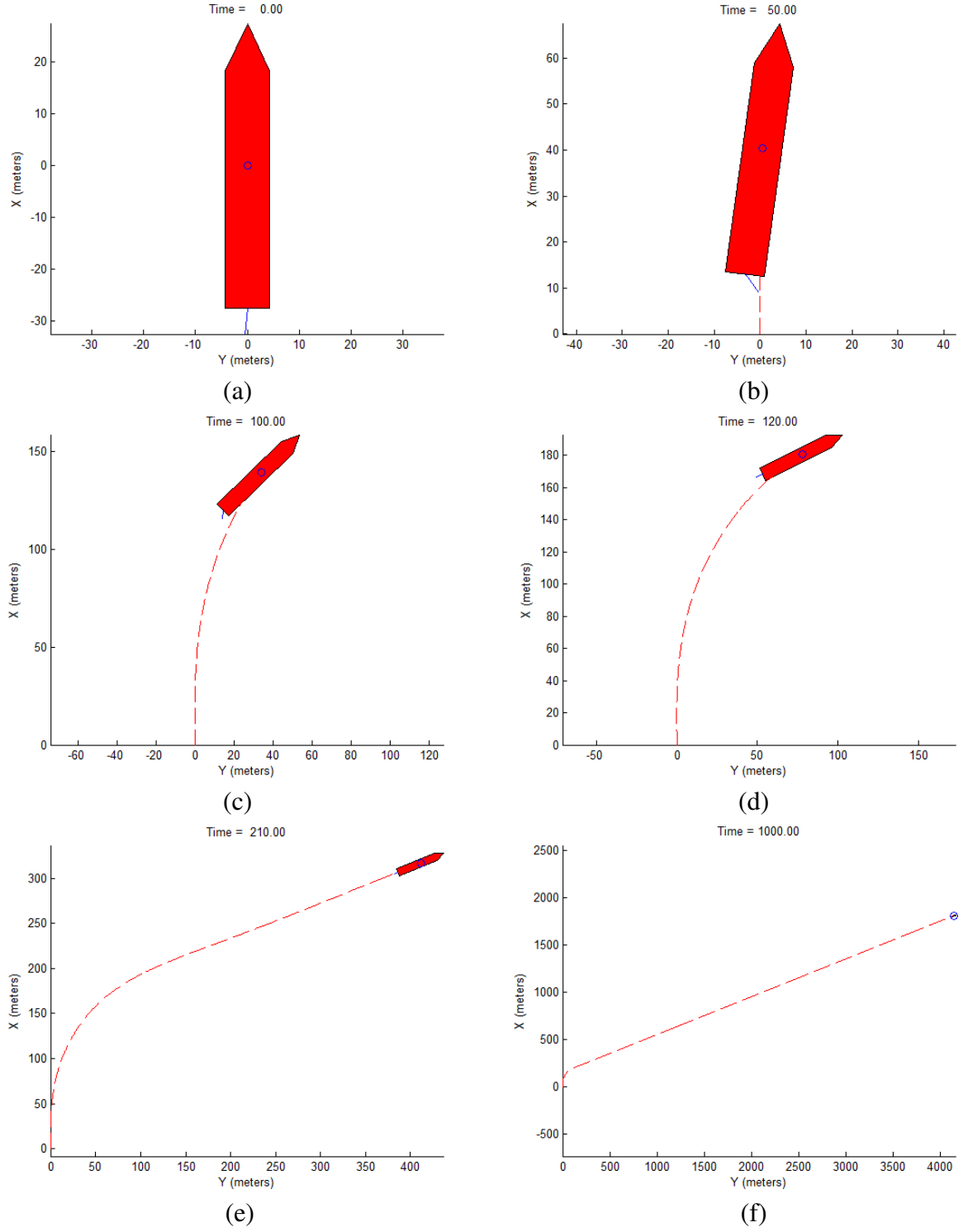


Figure 25: Trajectory Following Example

In Figure 26, we have plotted the vessel's X and Y components of the vessel's Global velocity. As can be seen here, the controller forces the velocity to converge on the desired values.

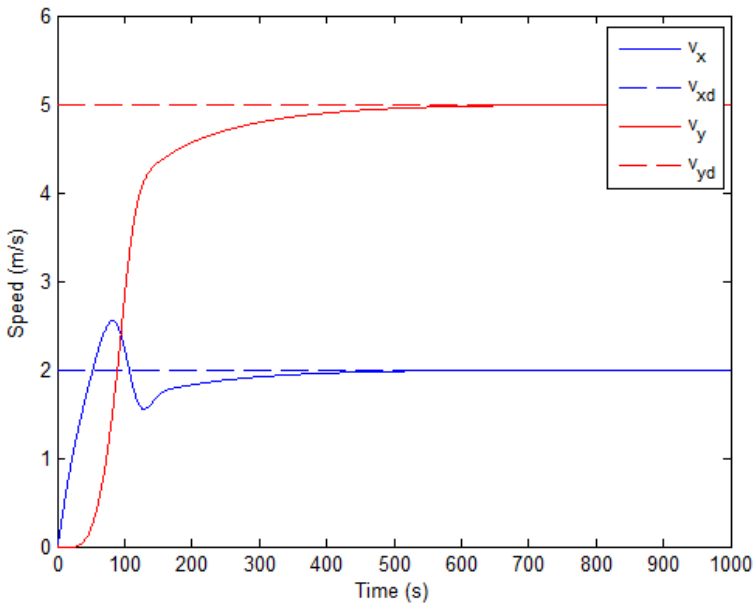


Figure 26: Global Frame Velocities vs. Time

The thrust and rudder controller gains for this simulation were:

$$k_{sp} = 5000, k_{si} = 200, k_r = -15$$

Next, we tested a controller this allowed us to follow a given point. This affords the vessel the capability of following pre-defined paths.

$$x_{cp} = x_{cg} + L_{cgi}\cos(\psi)$$
$$y_{cp} = y_{cg} + L_{cgi}\sin(\psi)$$
$$v_d^G = \begin{bmatrix}\dot{x}_d \\ \dot{y}_d\end{bmatrix} + k_f \begin{bmatrix}x_d - x_{cp} \\ y_d - y_{cp}\end{bmatrix}$$

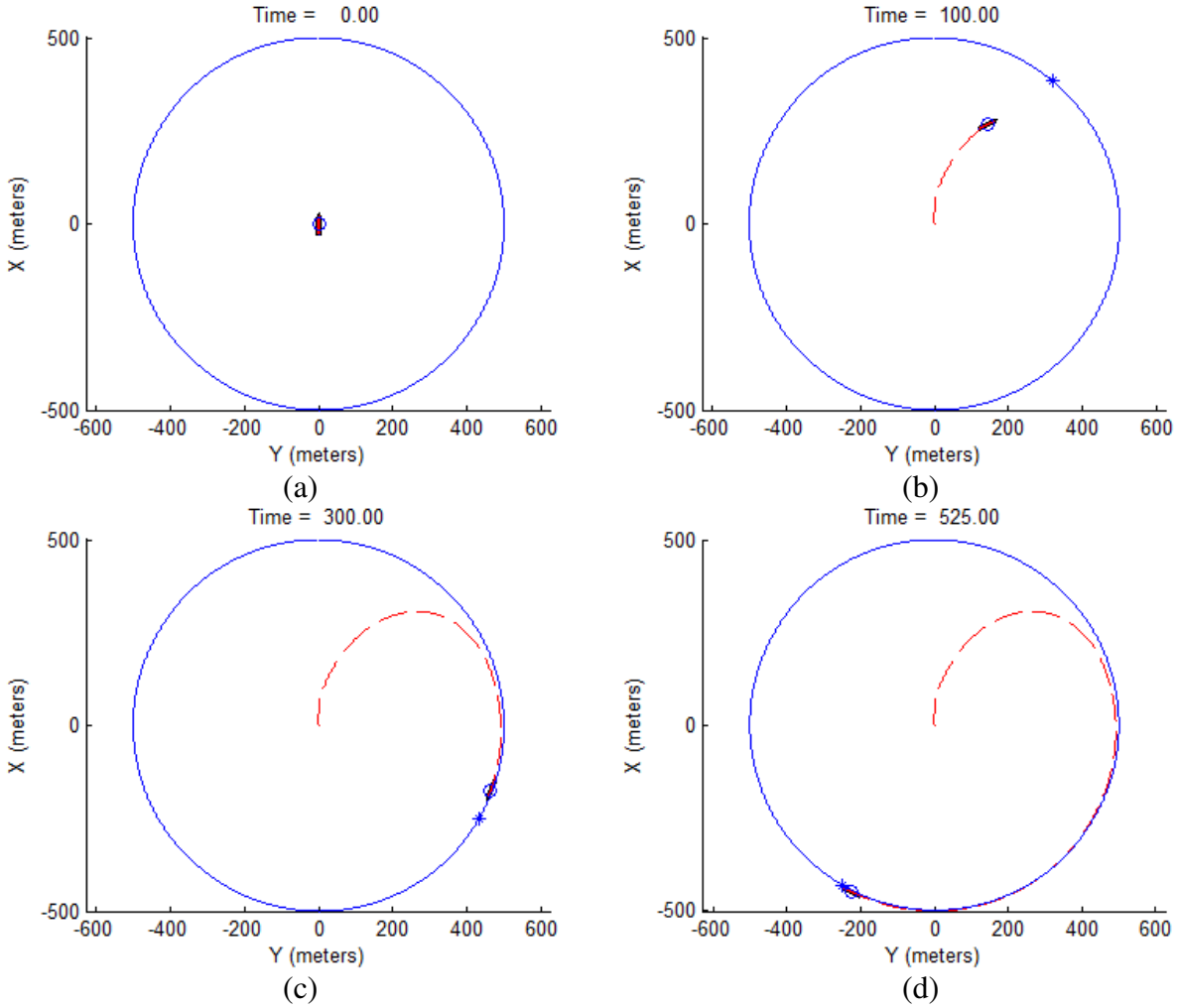
Where

- (x_{cp}, y_{cp}) is the location of the of the vessel's control point
- (x_{cg}, y_{cg}) is the location of the of the vessel's center of gravity
- L_{cgi} is the distance long the ship's surge axis between the control point and the center of gravity
- (x_d, y_d) is location of the position on which the vessel is attempting to converge
- $(\dot{x}_d, \dot{y}_d)$ is the Global velocity of the position on which the vessel is attempting to converge
- v_d^G is the desired velocity of the vehicle expressed in the global frame

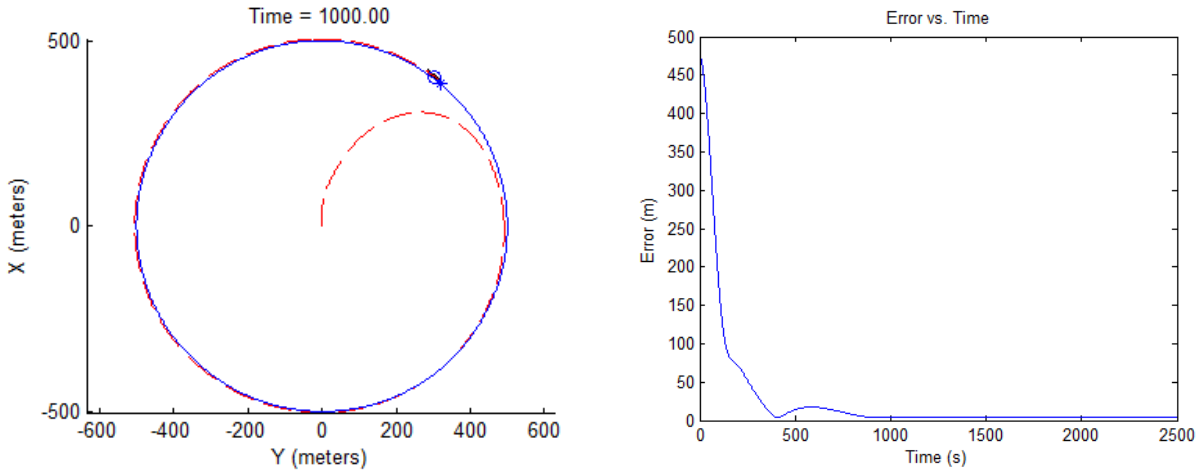
In the following simulation, we defined (x_d, y_d) as a time-varying value defined as:

$$x_d = 500cos\left(\frac{2\pi}{900}t\right)$$
$$y_d = 500sin\left(\frac{2\pi}{900}t\right)$$

The desired track that this produces is a circle with a radius of 500m. Additionally, the point will complete one full circle every 900s. The vessel starts from rest at location (0,0)m. In Figure 27(a), t =0s. The vessel is at rest with its center of gravity located at the origin of the Global reference frame. The point being tracked is located at the top of the circle at the coordinate (500,0)m. Figure 27(b) shows the vessel gains speed and turns to follow the tracking point. Figure 27(c) shows the vessel as is gets on the path of the tracking point but still lags behind it. In Figure 27(d), the vessel closes with the tracking point. Lastly, in Figure 27(e), we see that at the end of the simulation the vessel has remained in close contact with the tracking point. In Figure 27(f), we have a plot of the tracking error versus time. The steady-state error of the vessel was 3.54m.



(a) (b) (c) (d)



(e) (f)

Figure 27: Trajectory Following Simulation

The thrust and rudder controller gains for this simulation were:

$$k_{sp} = 1000, k_{si} = 5, k_r = -15$$

V. Swarm-Level Control

Once we had developed a model that allowed for the control of single vessels, we set out to develop a swarm-level controller capable of coordinating several vessels. The controller has a desired state to which the swarm should converge. Based on the actual swarm state at a given moment, the controller provides a desired velocity vector for each of the individual vessels in the swarm. This will cause the swarm to converge to the desired state. For a swarm with n members, the swarm state, q, is defined as:

$$q = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix}$$

Where (x_i, y_i) represents the location of the ith vessel relative to the global coordinate frame.

The particular swarm-level controller we used for this study is known as a redundant manipulator formulation. The strength of this type of control is that it allows for the managing of multiple tasks at one time. These tasks are divided into categories: primary tasks and secondary tasks. A given primary task, $V(q)$ is defined as a function of the swarm state. To describe how vessel motions affect the value of the primary task function, we calculate the Jacobian of the primary task function:

$$J_t = \left[\frac{\partial V(q)}{\partial q}\right] = \begin{bmatrix} \frac{\partial V(q)}{\partial x_1} \\ \frac{\partial V(q)}{\partial y_1} \\ \vdots \\ \frac{\partial V(q)}{\partial x_n} \\ \frac{\partial V(q)}{\partial y_n} \end{bmatrix}$$

These tasks are coordinated by the following control law:

$$\dot{q}^d = \dot{q}^d_{Primary} + (I - J_t^+ J_t) \sum k_{si} \dot{q}^d_{Secondary} \quad [13]$$

Where:

- $\dot{q}^d$ contains the desired swarm vessel velocities
- $\dot{q}^d_{Primary}$ contains the desired vessel velocities from the primary task controller
- $\dot{q}^d_{Secondary}$ contains the desired vessel velocities from the secondary task controllers
- I is the n x n identity matrix
- J_t is the Jacobian of the primary task function
- J_t^+ is the pseudoinverse of J_t

To determine how changes in value of the primary task function affect the desired vessel motions, we need to use the inverse of J_t. Because, the Jacobian is non-square, and therefore, non-invertable, the right-handed pseudoinverse of J_t^+ is used. The right-handed pseudoinverse of J_t is defined as $J_t^+ = \mathrm{J_t^T}(\mathrm{J_t J_t^T})^{-1}$. The term $(I - J_t^+ J_t)$ projects the desired vessel velocities of the secondary task controllers onto the null-space of the primary task Jacobian. This projection ensures that the secondary tasks have minimal effect on the primary task.

The desired velocities from the primary task controller are expressed as:

$$\dot{q}^d_{Primary} = J_t^+ \left(k_p\left(V(q)^d - V(q)\right)\right) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x_f}(t) \\ \dot{y_f}(t) \end{bmatrix}$$

Where:

- $V(q)^d$ represents the desired value of the Primary Task Function
- $V(q)$ represents the value of the Primary Task Function as a function of the swarm state
- $\left(\dot{x_f}(t), \dot{y_f}(t)\right)$ is the feed forward velocity of the primary task controller. This ensures that the swarm converges to the desired state.

The term $V(q)$ represents the swarm's primary task function. This function describes some primary condition that we may wish to satisfy. In this case, it will describe the desired formation of the swarm. For formation following involving a swarm of *n* vessels, $V(q)$ is defined as

$$V(q) = \sum_{i=1}^{n} U_i$$

Where U_i is determined by the desired formation shape, which will be covered below.

a. Circle

In our first case, the formation is the a circle of radius, *r*, with its center at (x_d, y_d). To cause the formation to move, the two terms are time varying. For motion in a straight path, they are defined by the following equations:

$$x_d = v_x t$$
$$y_d = v_t t$$

Formation following was accomplished through the use of an artificial potential field defined by the following equation:

$$U_i = ((x_i - x_d)^2 + (y_i - y_d)^2 - r^2)^2$$

Here, U_i represents the potential at the point (x_d, y_d). The potential of any point located on the aforementioned circle is zero. This has been illustrated in Figure 28. In the figure, a red circle has been plotted over the 3D surface to show the intended formation.

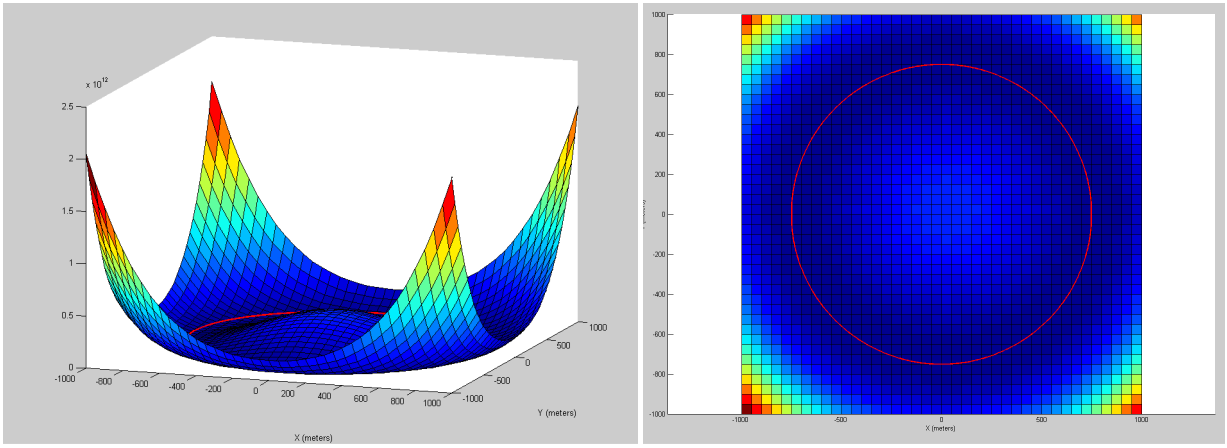


Figure 28: Artificial Potential Field for Circular Formation

To calculate the total value of the swarm function for a swarm with *n* members, we use the equation below:

$$V(q) = \sum_{i=1}^{n} U_i$$

For this particular swarm function, the Jacobian matrix is:

$$J(q) = \begin{bmatrix} 2(2x_1 - 2tv_x)\left((x_1 - tv_x)^2 + \left(y_1 - tv_y\right)^2 - r^2\right) \\ 2\left(2y_1 - 2tv_y\right)\left((x_1 - tv_x)^2 + \left(y_1 - tv_y\right)^2 - r^2\right) \\ \vdots \\ 2(2x_n - 2tv_x)\left((x_n - tv_x)^2 + \left(y_n - tv_y\right)^2 - r^2\right) \\ 2\left(2y_n - 2tv_y\right)\left((x_n - tv_x)^2 + \left(y_n - tv_y\right)^2 - r^2\right) \end{bmatrix}^T$$

Additionally, the pseudoinverse is defined by the equation $J^+ = J^T(JJ^T)^{-1}$. [13]

b. Ellipse

In our second case, we defined the formation of the swarm as an ellipse. An example has been provided in Figure 29.

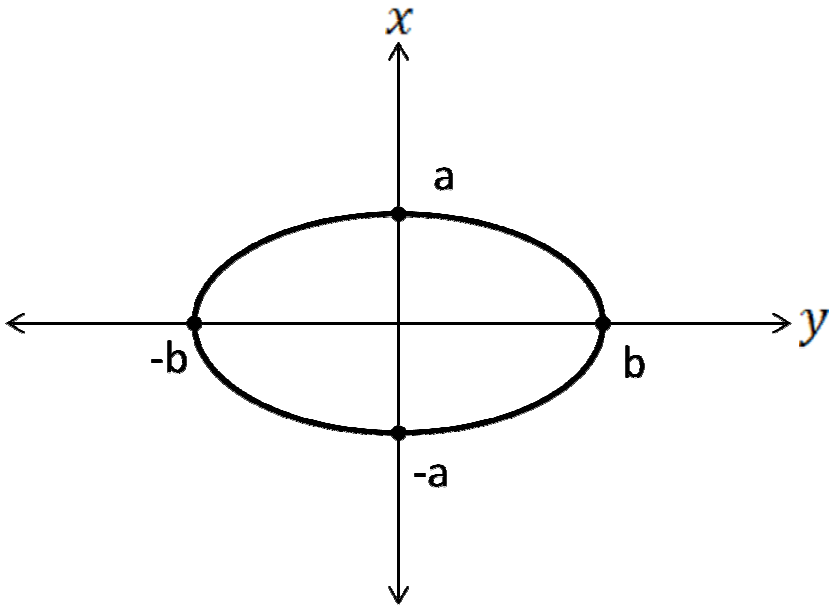


Figure 29: Basic Ellipse

This is again done using artificial potential fields. In this case, however, we define the potential field using:

$$U_i = \left(\frac{(x_i - x_d)^2}{a^2} + \frac{(y_i - y_d)^2}{b^2} - 1\right)^2$$

The center of the ellipse is time varying, with its coordinates defined as:

$$x_d = v_x t$$
$$y_d = v_t t$$

This creates an ellipse as shown in Figure 30, where *a* is the radius of the ellipse along the x-axis and *b* is the radius along the y-axis.

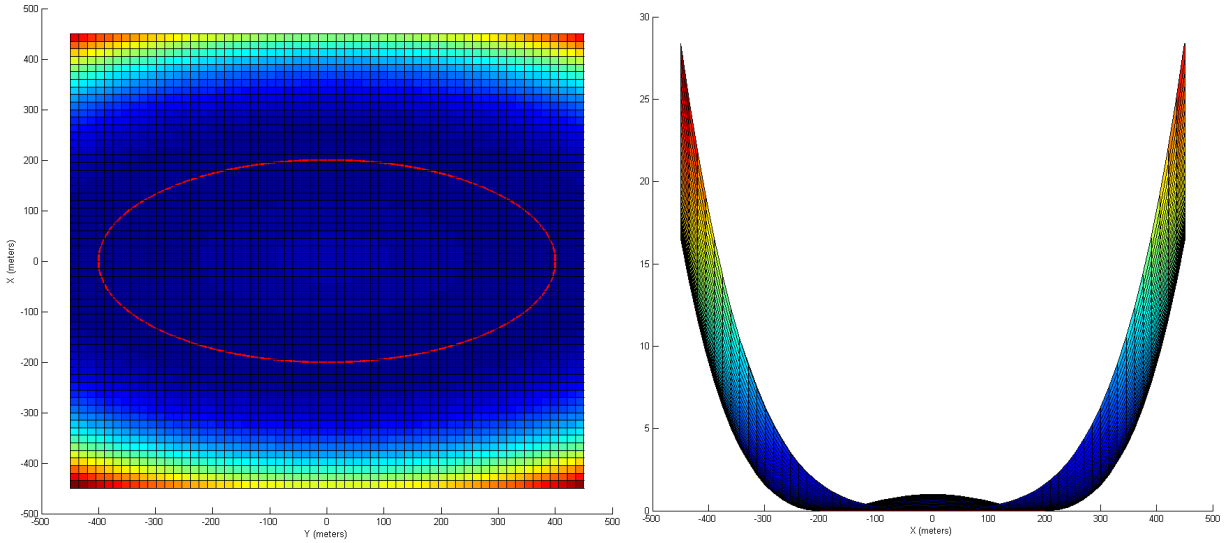


Figure 30: Artificial Potential Field for Elliptical Formation

To cause the formation to move, we designated

$$x_d = v_x t$$
$$y_d = v_t t$$

The value of the swarm function for a swarm of n vessels is again calculated according to the following equation:

$$V(q) = \sum_{i=1}^{n} U_i$$

c. Self-Avoidance

For movement in formation, it is important to ensure that the vessels do not collide. To ensure that separation is maintained, we define an avoidance vector with a magnitude defined below: [14]

$$M_{avoid} = \frac{D_{max} - D_i}{D_{max} - D_{min}}$$

Where D_{min} is the minimum distance at which the avoidance controller is active. D_{max} is the maximum distance at which the avoidance controller is active.

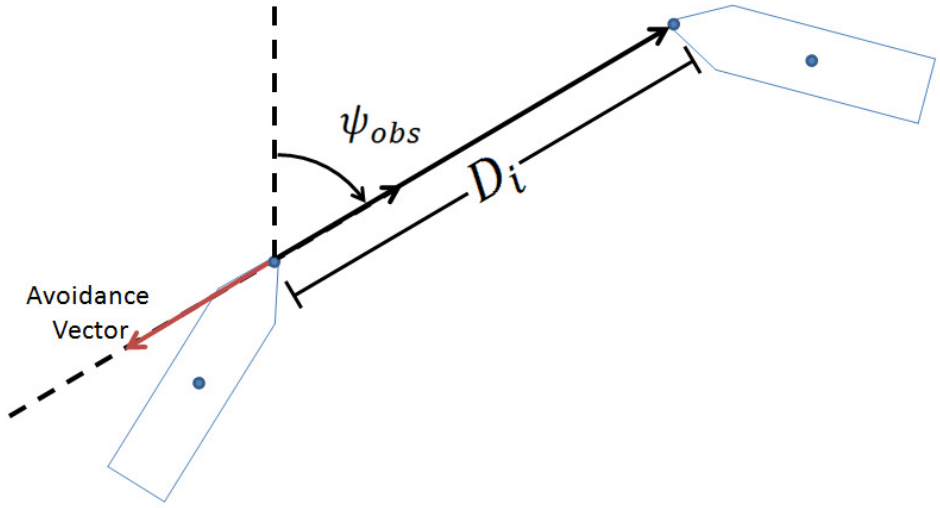


Figure 31: Illustration of Self-Avoidance Controller

In Figure 31, we show how the controller generates an avoidance vector for a vessel based on the distance and orientation of other vessels in the swarm. The equation for M_{avoid} generates an artificial potential field. This field is shown in Figure 32. The potential drops to 0 at D_{max} and continues at that value as the distance increases. This is because the avoidance controller automatically generates the zero vector for cases where D is greater than D_{max}.

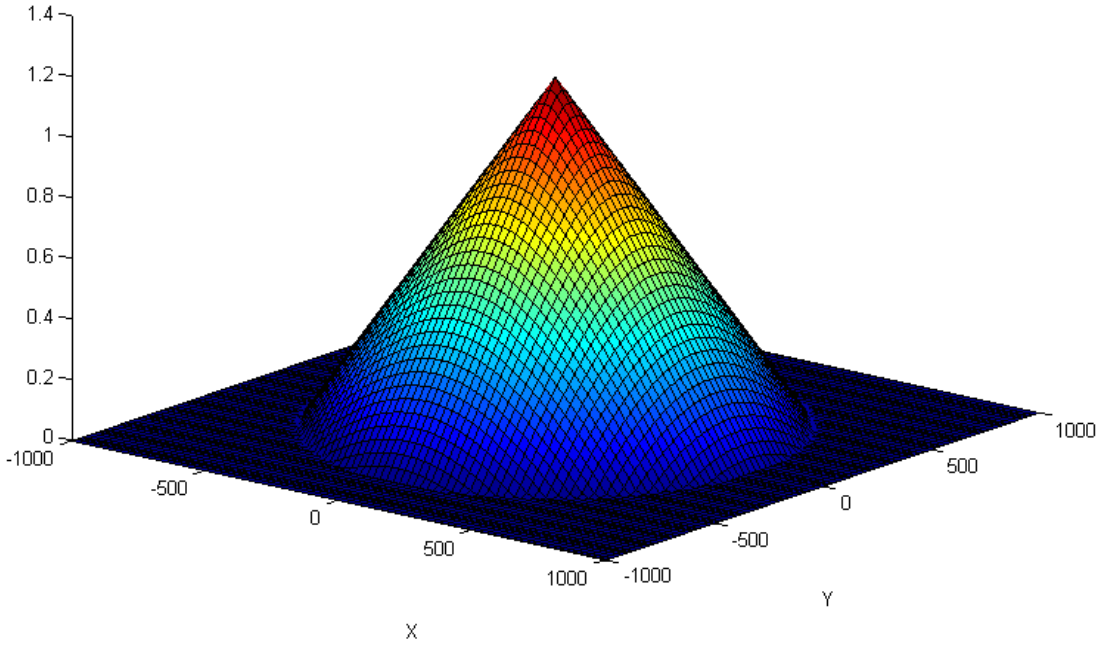


Figure 32: Artificial Potential Field Generated by Avoidance Control

The avoidance vector is defined by the following equation:

$$\dot{q}_{Avd} = M_{avoid}[\cos(\psi_{obs}) \quad \sin(\psi_{obs})]^T$$

In cases where there are multiple vessels that must be avoided, multiple avoidance vectors are calculated and summed.

d. Controller Tests

i. Circle

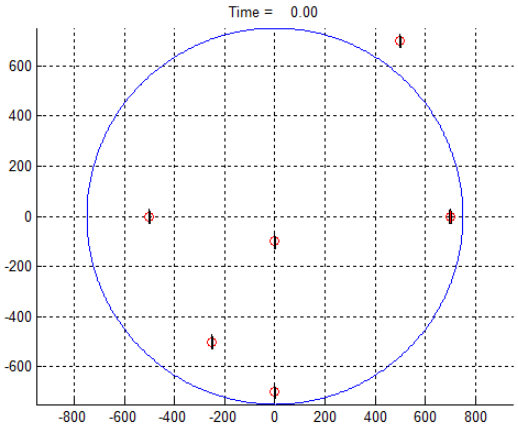
To demonstrate our swarm control method using a circular formation, a swarm of six vessels was created. The desired swarm velocity is 4m/s towards in a Northerly direction. In this simulation, the controller gains and formation parameters for the swarm-level controller were:

$$k_{primary} = .25, k_{avoidance} = 1, r = 750$$

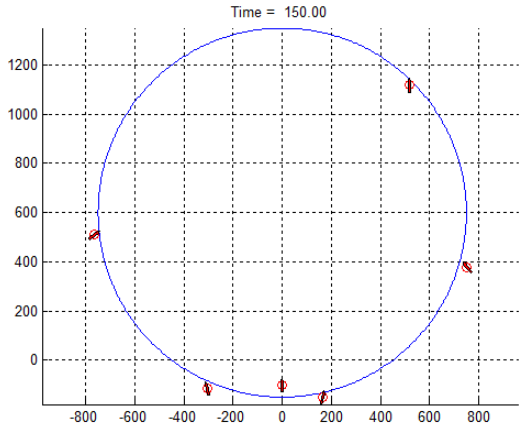
For the vessel's thrust and rudder system, the control gains were:

$$k_{sp} = 7500, k_{si} = 200, k_r = -15$$

In Figure 33(a), the vessels start from rest at arbitrary locations. In Figure 33(b) the vessels move onto formation. In Figure 33(c)-(e) the Self-Avoidance controller causes the vessels to spread out. To calculate the swarm error, we took the sum of each vessel's distance from the formation. In Figure 33(f), we plotted the error of the swarm throughout the simulation. The steady-state error in this simulation was 1.34m.



(a)



(b)

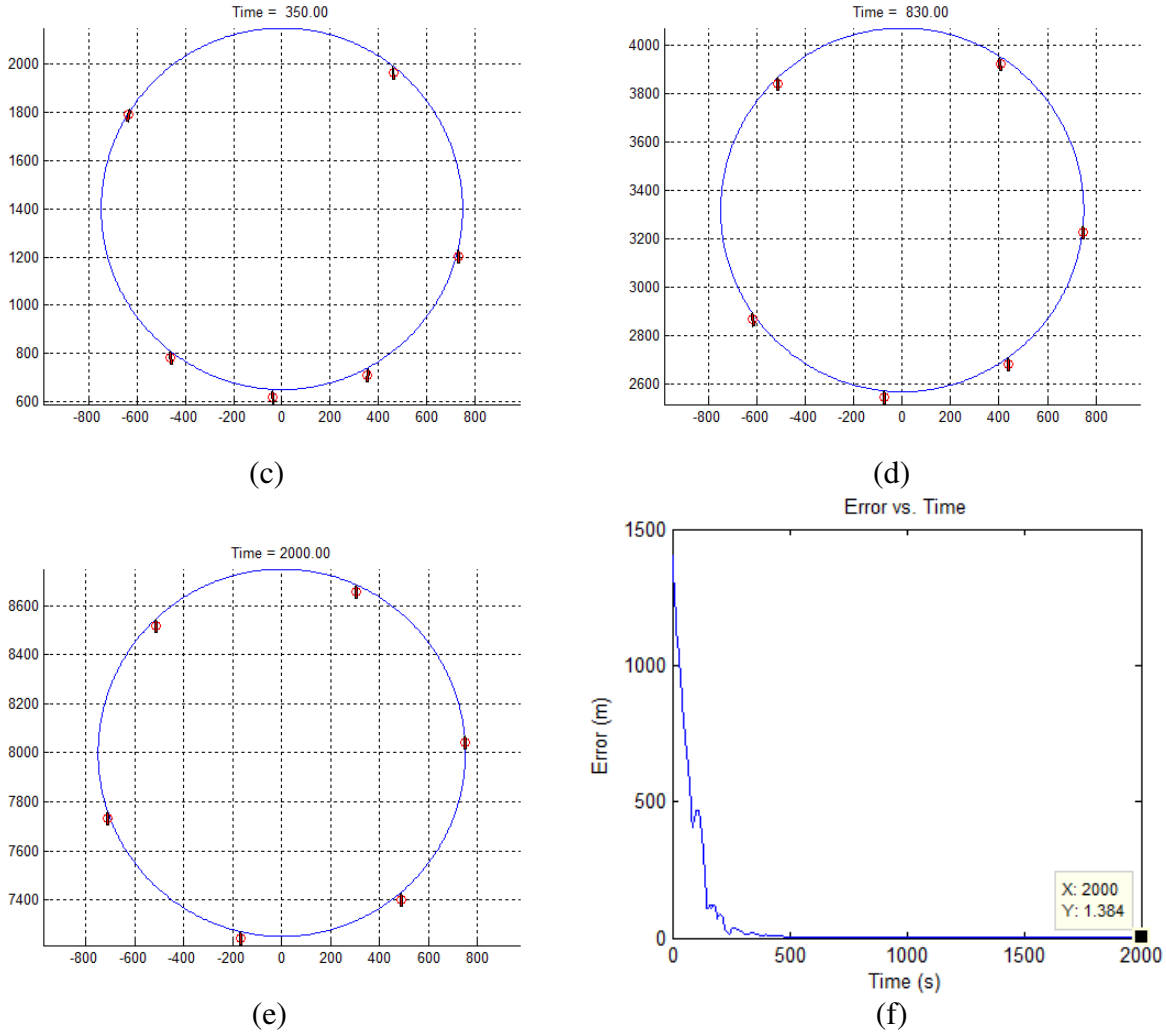


Figure 33: Circular Swarm Formation

ii. Ellipse

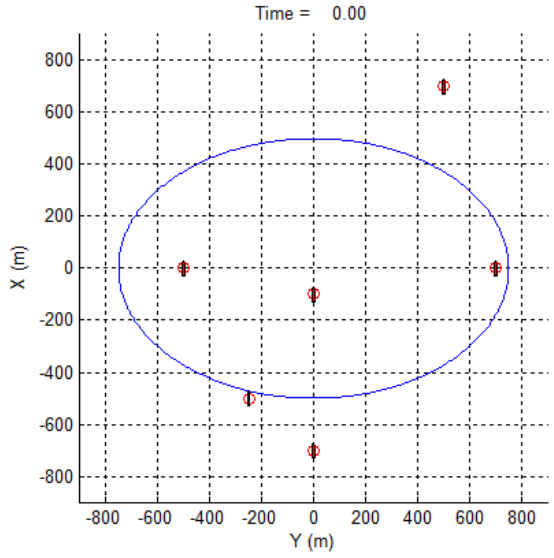
In this section we demonstrate an elliptical formation. The desired swarm velocity is 4m/s towards the North. In this simulation, the controller gains and formation parameters for the swarm-level controller were:

$$k_{primary} = .25, k_{avoidance} = 1, a = 500, b = 750$$

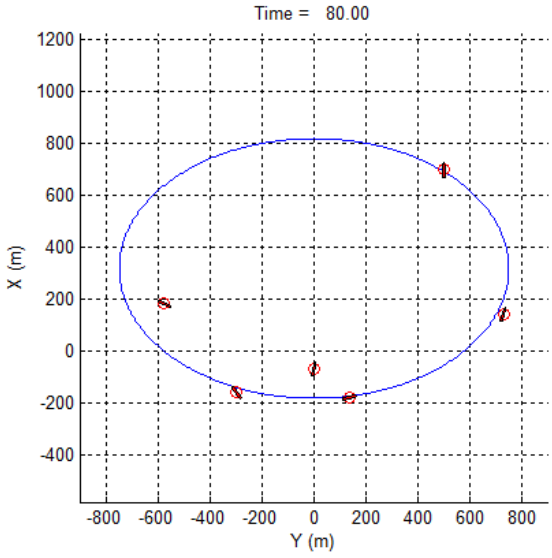
For the vessel's thrust and rudder system, the control gains were:

$$k_{sp} = 5000, k_{si} = 200, k_r = -15$$

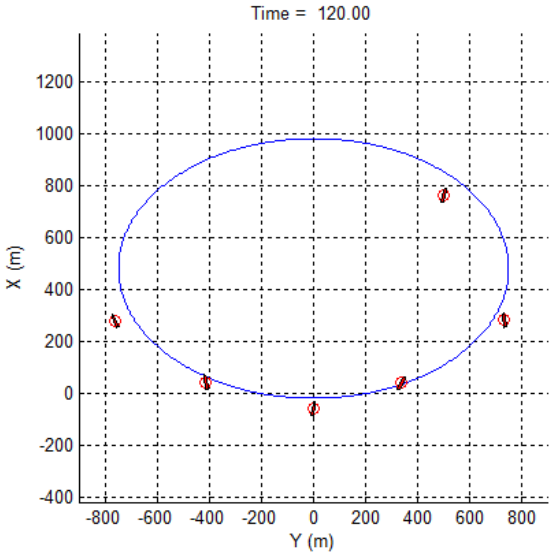
In Figure 34(a), the vessels start from rest at arbitrary locations. In Figure 34(b)-(c) the vessels move onto the formation. In Figure 34(d)-(e) the Self-Avoidance controller causes the vessels to spread out. For this simulation, we calculated the swarm error using the same method as in the previous simulation. In Figure 34(f), we plotted the error of the swarm throughout the simulation. The steady-state error in this simulation was 6.45m.


Time = 0.00
X (m)
Y (m)
800
600
400
200
0
-200
-400
-600
-800
-800 -600 -400 -200 0 200 400 600 800

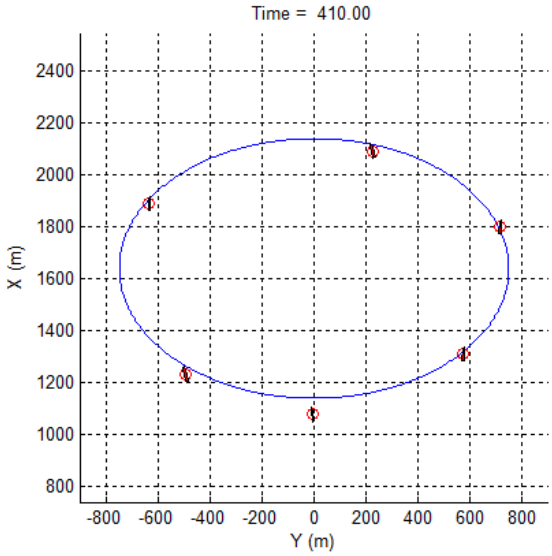
(a)


Time = 80.00
X (m)
Y (m)
1200
1000
800
600
400
200
0
-200
-400
-800 -600 -400 -200 0 200 400 600 800

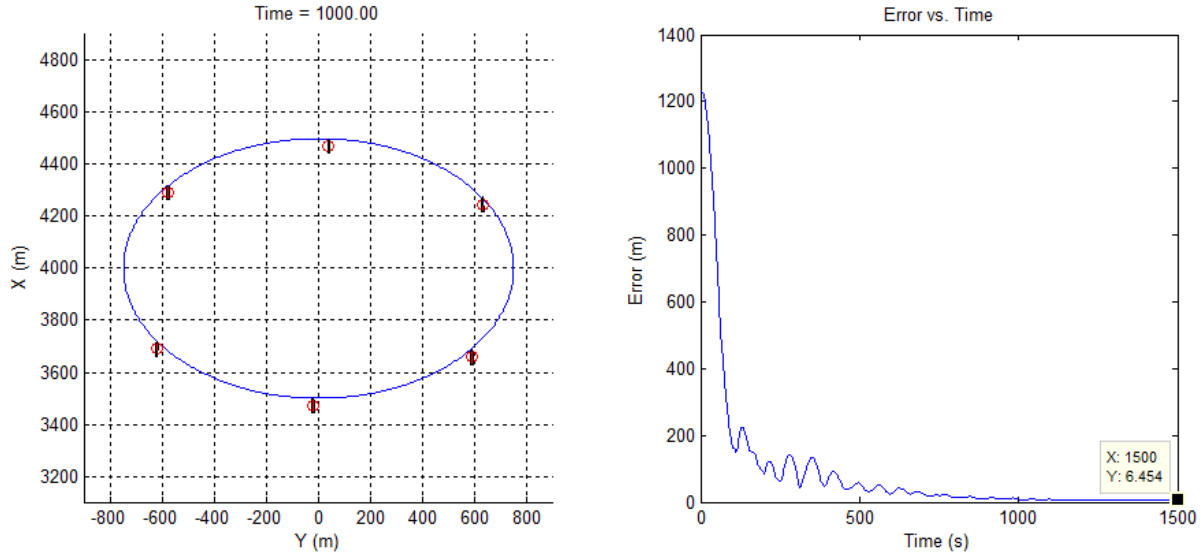
(b)


Time = 120.00
X (m)
Y (m)
1200
1000
800
600
400
200
0
-200
-400
-800 -600 -400 -200 0 200 400 600 800

(c)


Time = 410.00
X (m)
Y (m)
2400
2200
2000
1800
1600
1400
1200
1000
800
-800 -600 -400 -200 0 200 400 600 800

(d)



(e) (f)

Figure 34: Elliptical Swarm Formation

VI. Disturbance Effects

This section contains the results from simulation in which we analyzed the effects of winds and currents on the swarm's ability to maneuver. The control gains and formation parameters in these simulations were the same as those used in Section V(d) for the for the circle formation following example.

a. Current

Of all the environmental effects, current is most disruptive. In this model, we assumed that the vessel moved with the current. To integrate current effects, the velocity of the current was added directly to the vessel's velocity.

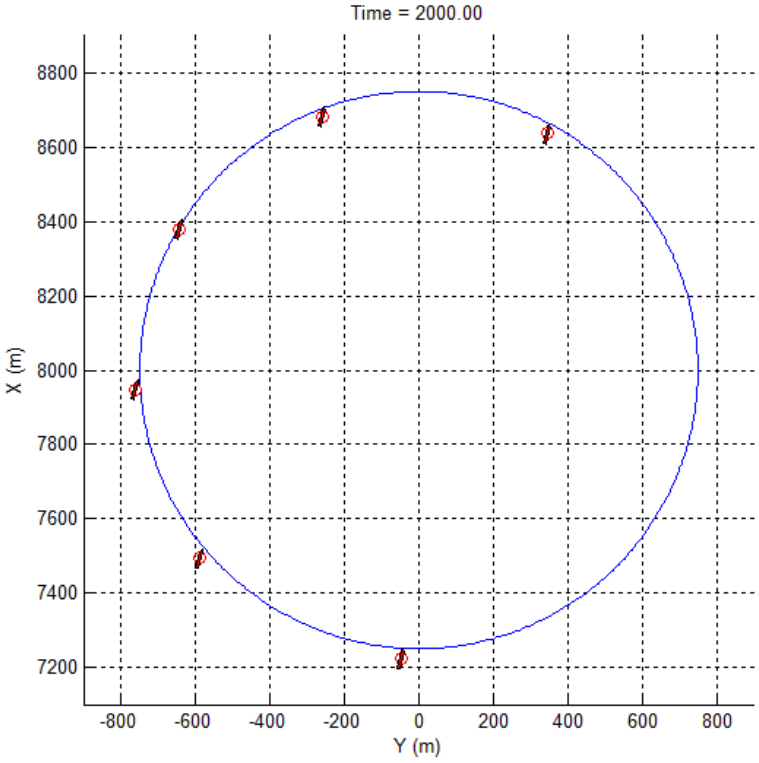


Figure 35: Circular Swarm Formation without Disturbance Compensation

To illustrate the effect of current on the swarm, we had the swarm attempt to move in formation at 4m/s in a direction of 000°. We subjected the swarm to a current moving at 1m/s in a direction of 270°. As can be seen in Figure 35, the current pushed the vessels to the Western side of the circle. This left a large gap in the eastern side of the swarm's formation. This also caused in increased in the steady-state swarm error. A plot of swarm error versus time for this simulation can be found below in Figure 36. As can be seen here, the steady-state swarm error increased to 34.4m.

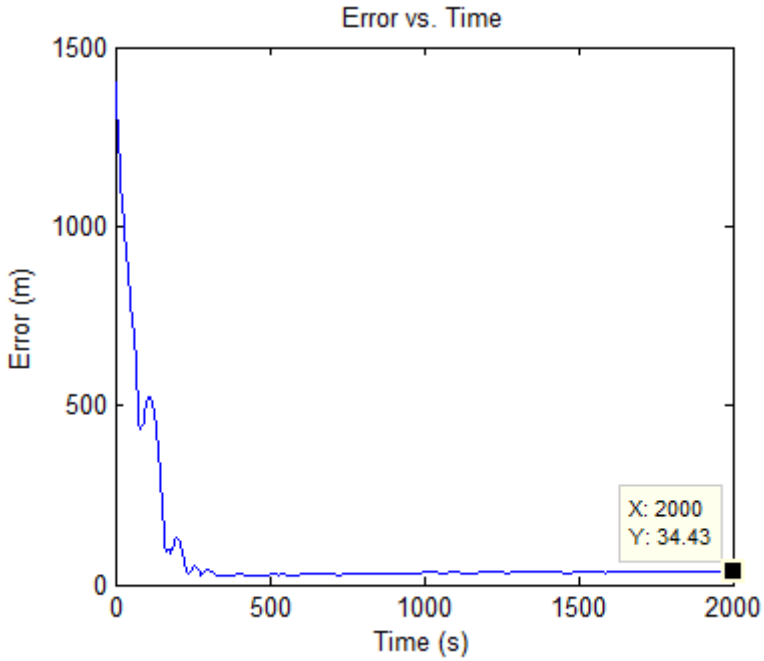
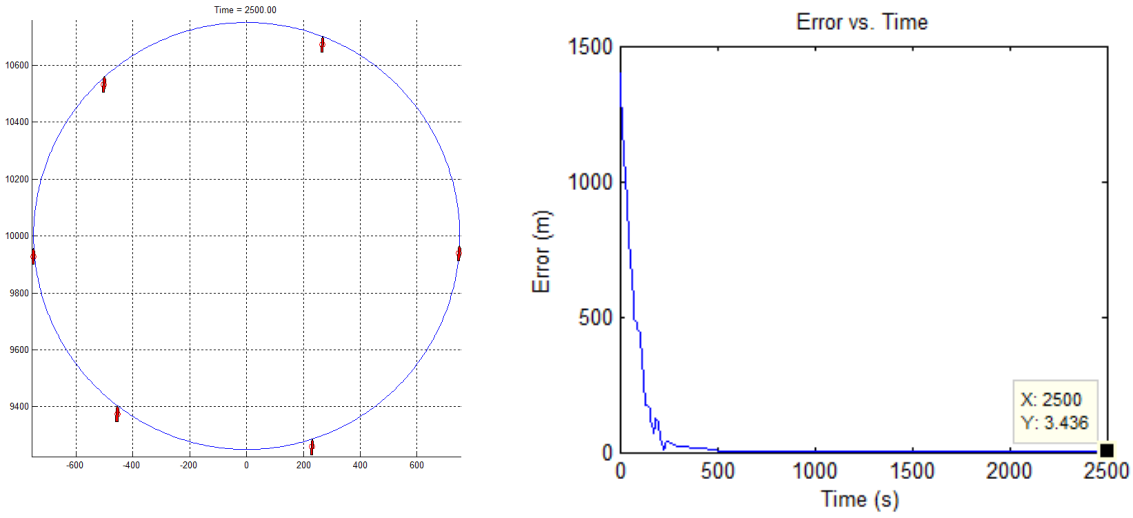


Figure 36: Error vs Time for Circular Swarm Formation With Current Effects

b. Wind

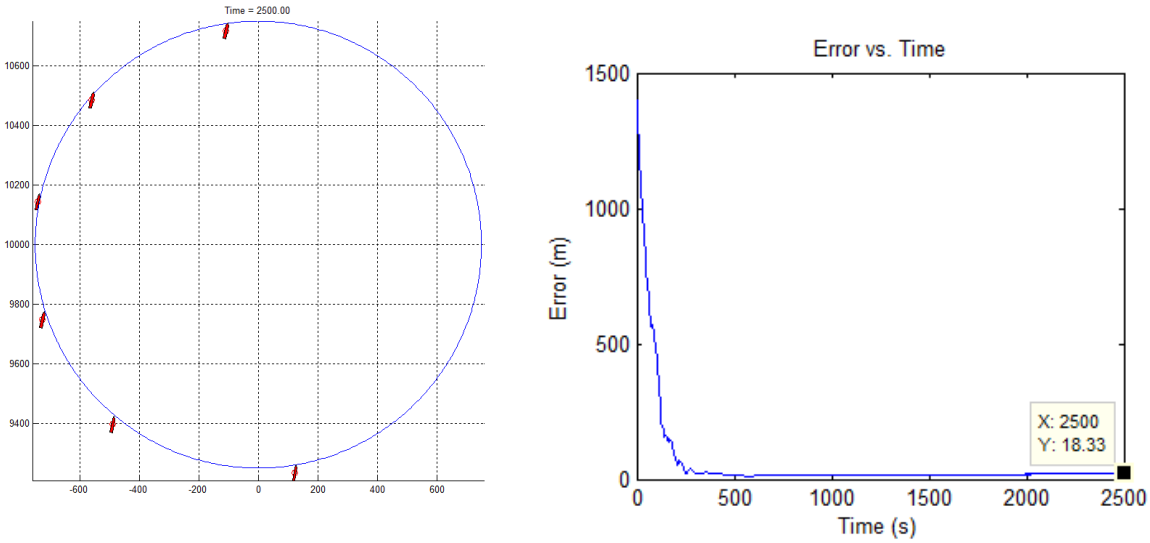
Using the same control gains as above, we conducted a simulation to test the effects of wind on the swarm formation. In the first test, we subject the swarm to a 10m/s wind coming from a direction of 090°. In Figure 37(a), we plotted the positions of the vessels at the end of the simulation. As seen here, five of the vessels have moved slightly farther to the western side of the formation than was observed with no wind (Figure 35). The disturbance also increased the steady-state error of the swarm to 3.44m.



(a) (b)

Figure 37: Swarm Formation with 10 m/s Wind from 090°

In the second test, we increased the speed of the wind to 20m/s. In Figure 38(a) we plotted the positions of the vessels at the end of the simulation. As seen here, the wind forced the vessel towards the western side of the formation. The steady-state error of the swarm had also increased to 18.33m.



(a) (b)

Figure 38: Swarm Formation with 20 m/s Wind from 090°

VII. Disturbance Compensation

In order to compensate for environmental disturbances, we created an approach called Curvature Optimization. This controller attempts to force the swarm members to locations on the formation where the disturbances have a lesser effect. For the following, the disturbance tested will be current, as this is the disturbance with the most disruptive effect on vessel maneuvering. First, we calculate the angle of encounter between the swarm and the disturbance, σ:

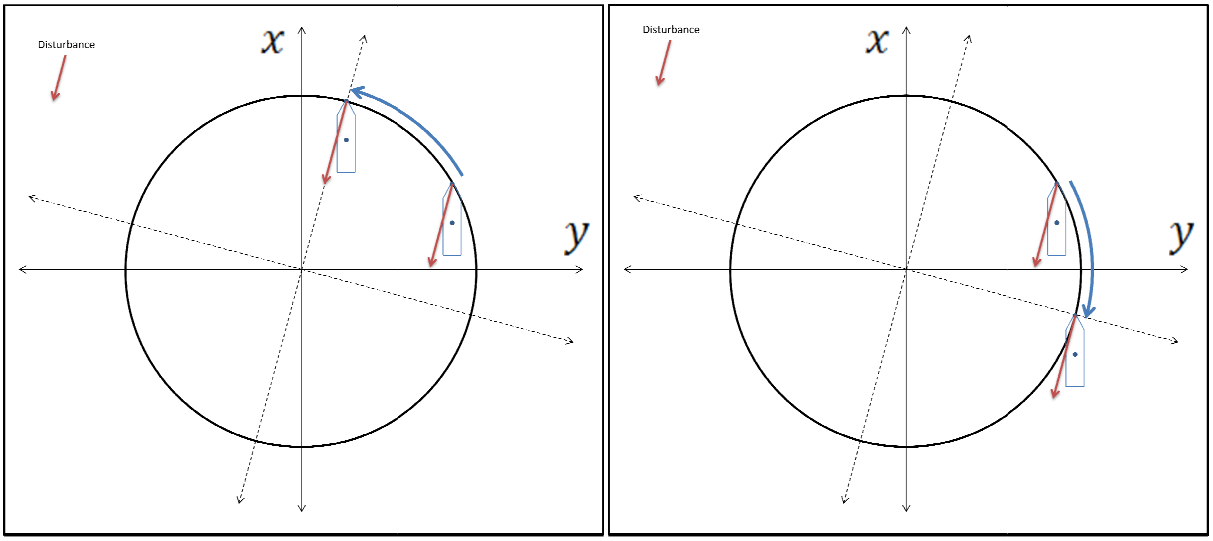
$$\sigma = \arctan\left(\frac{v_{cy} - v_y}{v_{cx} - v_x}\right)$$

Where (v_{cx}, v_{cy}) is the velocity of the disturbance and (v_x, v_y) is the desired velocity of the swarm. The swarm determines where on the formation a vessel should move based on the angle between the gradient vector of the artificial potential field at the location of the vessel and the disturbance encounter angle. Locations on the artificial potential field are local minima. To eliminate local effects, we modified the swarm function by taking the square root before calculating the gradient. The resulting equation for a circular formation is:

$$\nabla\left(\sqrt{U_i(x_i, y_i)}\right) = [2\mathrm{x_i} - 2\mathrm{x_d} \quad 2\mathrm{y_i} - 2\mathrm{y_d}]$$

Where (x_i, y_i) is the location of a vessel in the swarm and $(\mathrm{x_d}, \mathrm{y_d})$ is the location of the instantaneous center of the formation.

For this study, we tested two separate approaches to Curvature Optimization. Using Method 1, the swarm controller would attempt to force vessels to reach a location on the swarm where the gradient vector of the swarm function was perpendicular to the disturbance. Using Method 2, the swarm controller would attempt to force vessels to reach a location on the swarm where the gradient vector of the swarm function was parallel with the disturbance. Both of these methods are illustrated in Figure 39.



(a) (b)

Figure 39: Illustration of (a) Method 1 and of (b) Method 2

From the gradient of the swarm function, we can determine $\psi_{Gradient}$, using trigonometry. As seen in Figure 40, we can also determine the orientation of the lines tangent to the curve by either adding or subtracting 90° from $\psi_{Gradient}$. The controller will cause a vessel to move in the direction of either of two tangent vectors depending on the value of $\psi_{Gradient} - \sigma$.

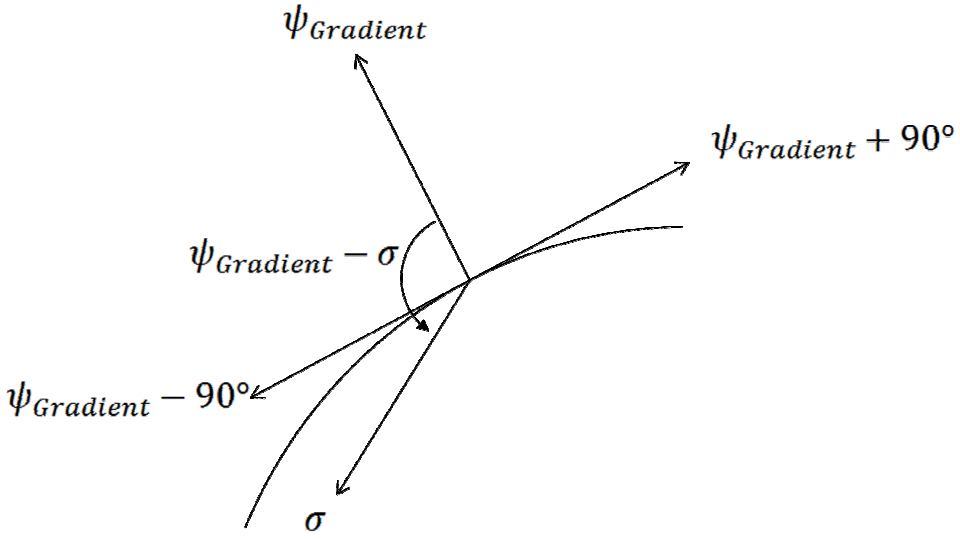


Figure 40: Gradient and Tangent Vectors

Using Method 1, the vessel would move in the direction of $\psi_{Gradient} - 90°$ if $\psi_{Gradient} - \sigma$ was between 0° and −90° or between 90° and 180°. Conversely, the vessel would move in the direction of $\psi_{Gradient} + 90°$ if $\psi_{Gradient} - \sigma$ was between −90° and −180° or between 0° and 90°.

Using Method 2, the vessel would move in the direction of $\psi_{Gradient} + 90°$ if $\psi_{Gradient} - \sigma$ was between 0° and −90° or between 90° and 180°. Conversely, the vessel would move in the direction of $\psi_{Gradient} - 90°$ if $\psi_{Gradient} - \sigma$ was between −90° and −180° or between 0° and 90°.

Both of the above methods will force members of the swarm to regions on the formation where the effects of the disturbance are less disruptive. To maximize the size of these regions, we decided to use an elliptical formation, this time rotated by an angle of σ as in Figure 41.

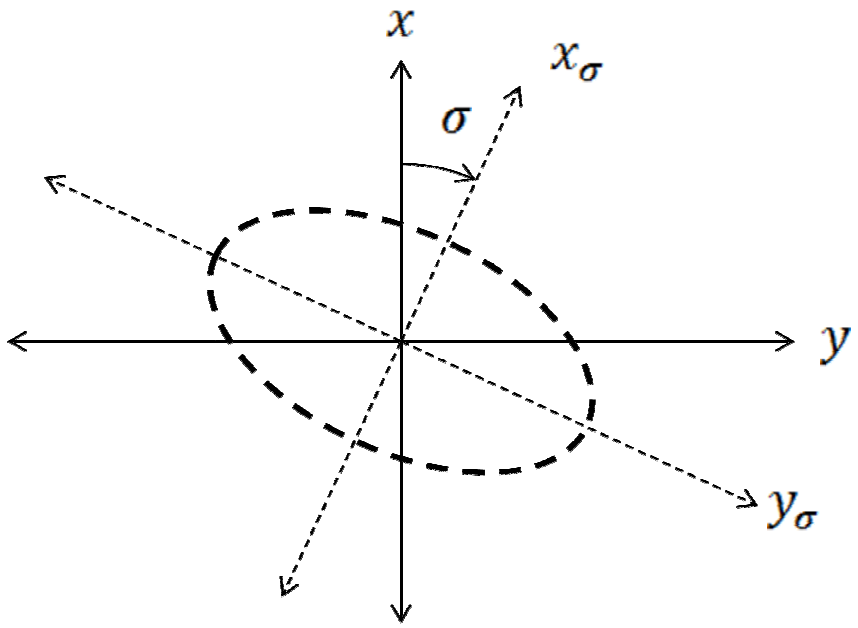


Figure 41: Rotation of Elliptical Formation

In order to rotate the ellipse as in the above graph, we have to redefine the swarm function. To produce an artificial potential field rotated by the angle, σ, we will define the vessel's location with respect to the artificial potential field as:

$$\begin{bmatrix} x_{\sigma i} \\ y_{\sigma i} \end{bmatrix} = \begin{bmatrix} \cos(\sigma) & \sin(\sigma) \\ -\sin(\sigma) & \cos(\sigma) \end{bmatrix} \begin{bmatrix} x_i - v_x t \\ y_i - v_t t \end{bmatrix} \quad [13]$$

Where (x_i, y_i) is the location of the *i*-th vessel relative to the Global frame. The potential at this location is:

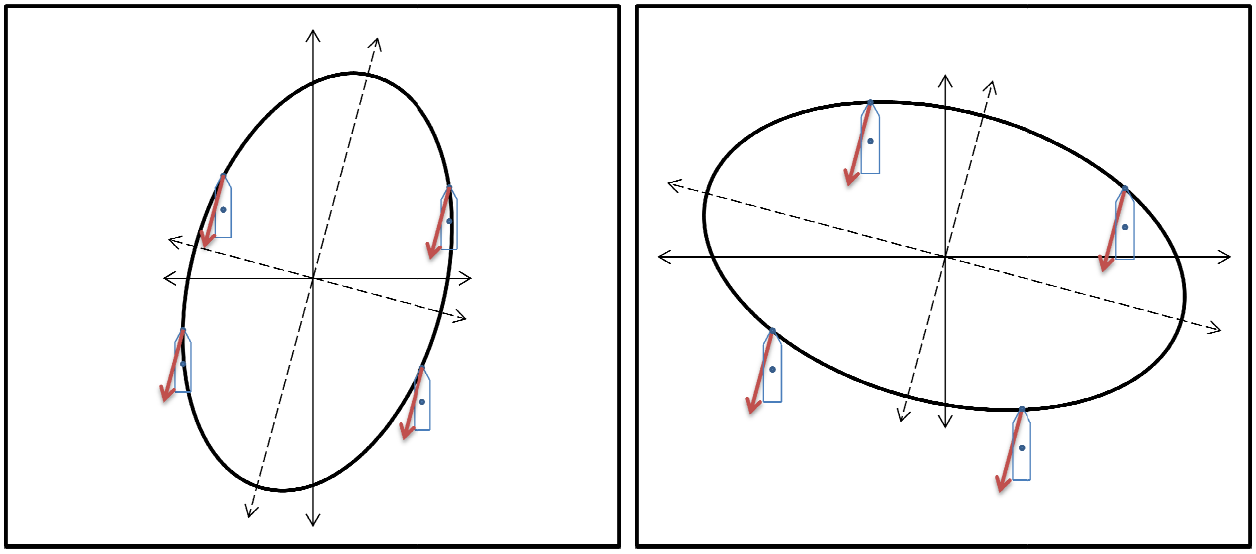
$$U_{\sigma i} = \left(\frac{x_{\sigma i}^2}{a^2} + \frac{y_{\sigma i}^2}{b^2} - 1 \right)^2$$

For the purposed of this controller, we will have to be able to determine the gradient vector of the artificial potential field at a given point. This is given by the equation below:

$$\nabla \left(\sqrt{U_i(x_i, y_i)} \right)$$

$$= \begin{bmatrix} \frac{2cos(\sigma)\big(cos(\sigma)(x_i - x_d) + sin(\sigma)(y_i - y_d)\big)}{a^2} - \frac{2sin(\sigma)\big(cos(\sigma)(y_i - y_d) - sin(\sigma)(x_i - x_d)\big)}{b^2} \\ \frac{2cos(\sigma)\big(cos(\sigma)(y_i - y_d) - sin(\sigma)(x_i - x_d)\big)}{b^2} + \frac{2sin(\sigma)\big(cos(\sigma)(x_i - x_d) + sin(\sigma)(y_i - y_d)\big)}{a^2} \end{bmatrix}^T$$

For Methods 1 and 2, the angle of rotation will be the same. The values of *a* and *b*, however will have to change. Method 1 demands a formation in which *a* is greater than *b*. For Method 2, the opposite is true. In Figure 42, we provide a general image of how the swarms are expected to settle in either case.



(a) (b)

Figure 42: Rotated Elliptical Formations Complimenting (a) Method 1 and (b) Method 2

VIII. Results

To determine the efficacy of our Disturbance Compensation controllers, we tested them on a simulated swarm of vessels attempting to move in formation at 4m/s in a northern direction while encountering environmental effects. In this simulation, the controller gains for the swarm-level controller were:

$$k_{primary} = .25,\ k_{avoidance} = 7.5, k_{Curvature\ Optimization} = 10$$

The self-avoidance controller gain was increased in these tests to allow for self-avoidance despite conflict with the Curvature Optimization controller. For the simulation using no disturbance correction, the formation parameters were:

$$a = 750,\ b = 750$$

For the simulation using Method 1, the formation parameters were:

$$a = 500,\ b = 750$$

For the simulation using Method 2, the formation parameters were:

$$a = 750,\ b = 500$$

For the vessel's thrust and rudder systems, the control gains were:

$$k_{sp} = 10000,\ k_{si} = 200,\ k_r = -15$$

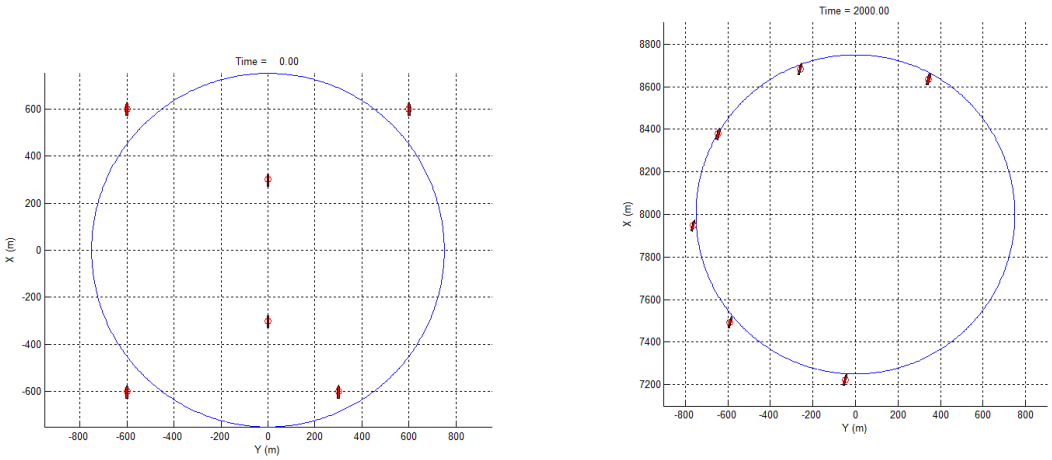
a. Current

The first test subjected the swarm to a 1m/s current moving in the direction of 270°. We used the disturbance angle equation to determine σ:

$$\sigma = \arctan\left(\frac{v_{cy} - v_y}{v_{cx} - v_x}\right)$$

Where the swarm velocity vector was $(-4,0)$ m/s and the current velocity vector was $(0,-1)$ m/s In this case $\sigma = -165°$

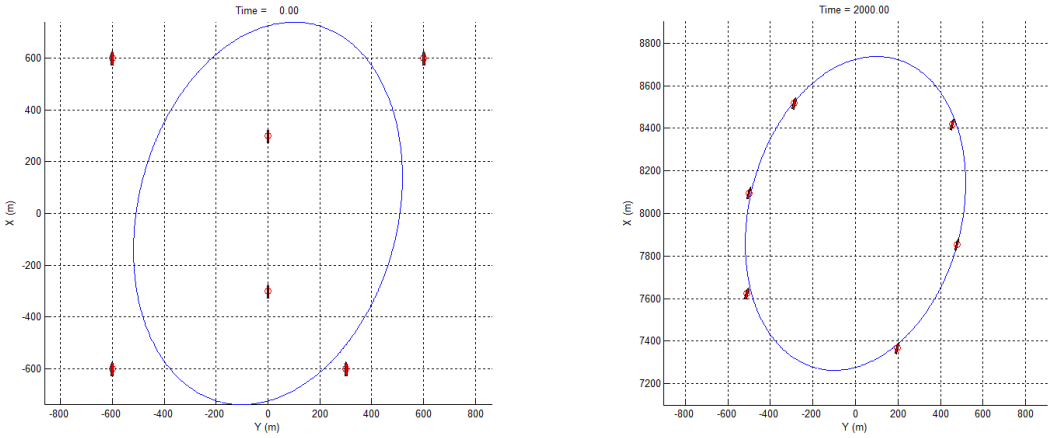
As our control group, we first tested an elliptical formation with equal values of *a* and *b*, effectively making a circle. Additionally, we had Curvature Optimization turned off for this test. As can be seen in Figure 43, the vessels start by spreading out and off the formation, then end on the formation. As noted earlier in the paper, the disturbance forced the vessels towards the western part of the formation.



(a) (b)

Figure 43: (a) Start and (b) End of Simulation with Circular Formation and no Curvature Optimization

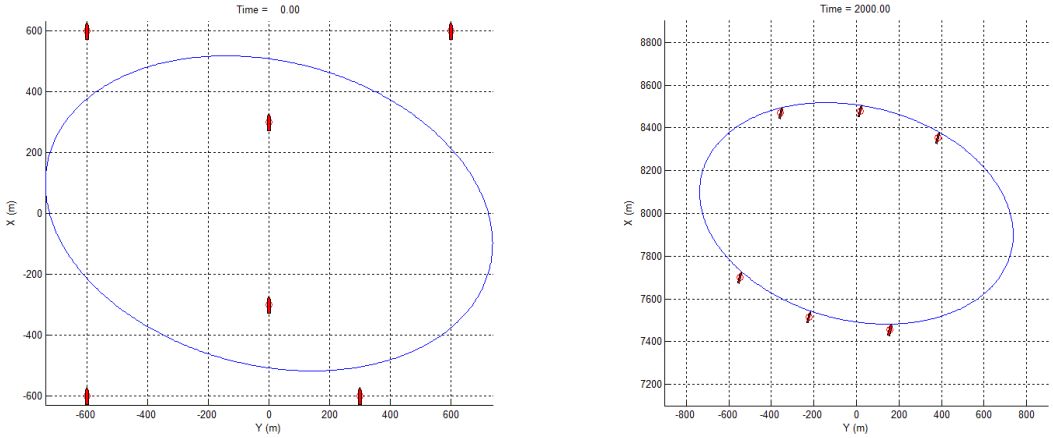
Next, we tested Method 1. As can be seen in Figure 44, the vessels started out in the same positions as in the previous test. By the end of the simulation, they have been able to get in formation and have moved into the regions commanded by the Curvature Optimization controller.



(a) (b)

Figure 44: (a) Start and (b) End of Simulation with Method 1 of Curvature Optimization and Complimentary Elliptical Formation

Last, we tested Method 2. As can be seen in Figure 45, the vessel's started out in the same positions as in the previous tests. By the end of the simulation, they have been able to get in formation and have moved into the regions commanded by the Curvature Optimization controller.



(a) (b)

Figure 45: (a) Start and (b) End of Simulation with Method 2 of Curvature Optimization and Complimentary Elliptical Formation

From just the views of the vessels in formation, we can tell that both Methods 1 and 2 resulted in improved dispersion on the circle compared to the circle with no disturbance compensation. To compare Methods 1 and 2, we compared the total error of the swarm over the course of the simulations. As a baseline, we found that at the end of the simulation using no Disturbance Correction, the swarm had a total error of 24.7m. Method 1, surprisingly, increased the Total Error to 32.7m. Method 2, on the other hand, caused a reduction in total error to 7.98m. This amounts to a 75.6% drop in the swarm's total error.

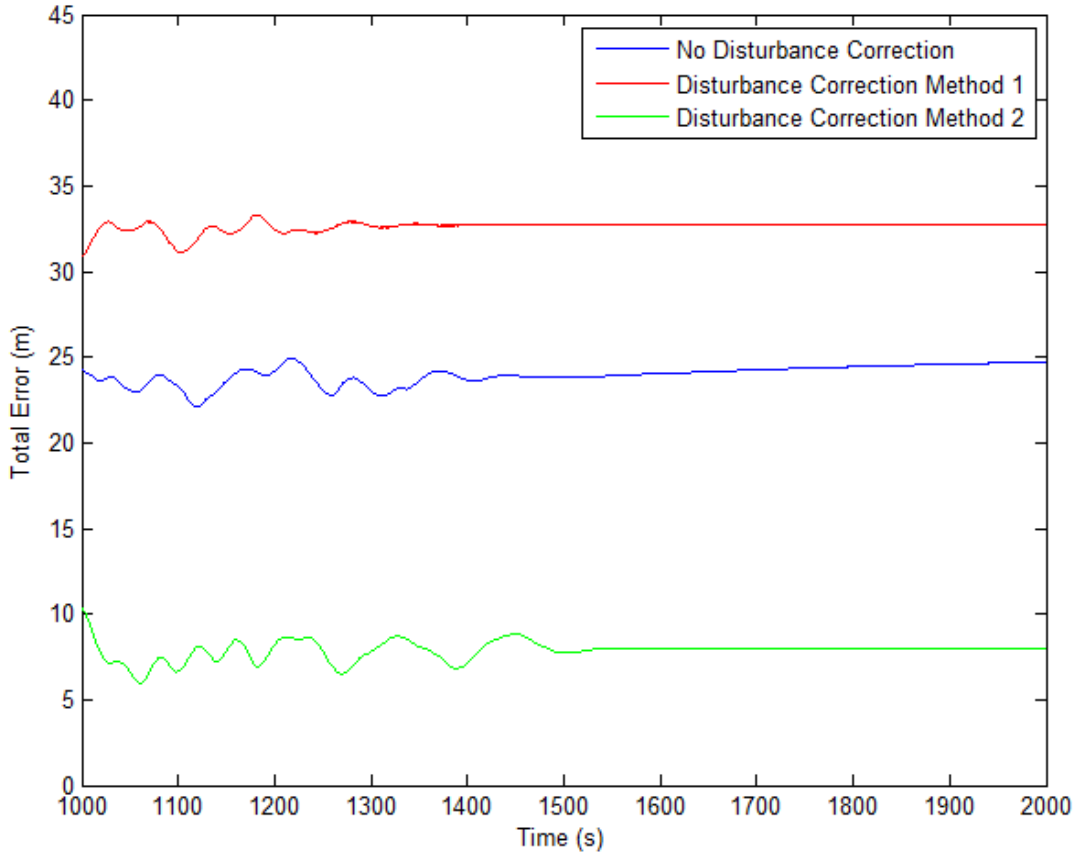


Figure 46: Total Error versus Time for Various Tests

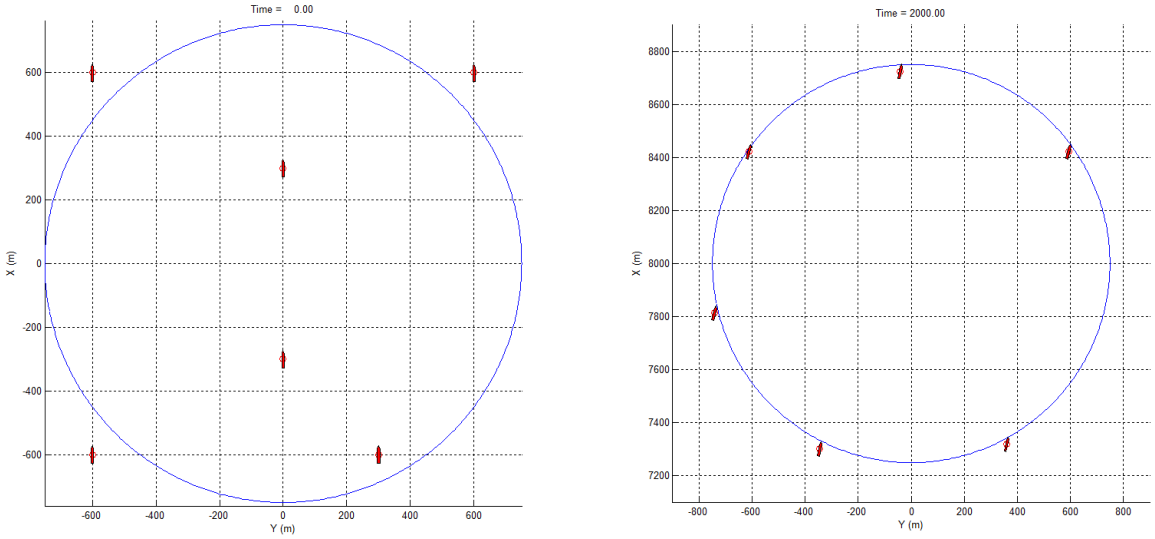
b. Wind

The second test subjected the swarm to a 20m/s wind moving from the direction of 090°. We used the disturbance angle equation to determine σ:

$$\sigma = \arctan\left(\frac{v_{cy} - v_y}{v_{cx} - v_x}\right)$$

Where the swarm velocity vector was $(-4,0)$ m/s and the wind velocity vector was $(0,-20)$ m/s In this case $\sigma = -101°$.

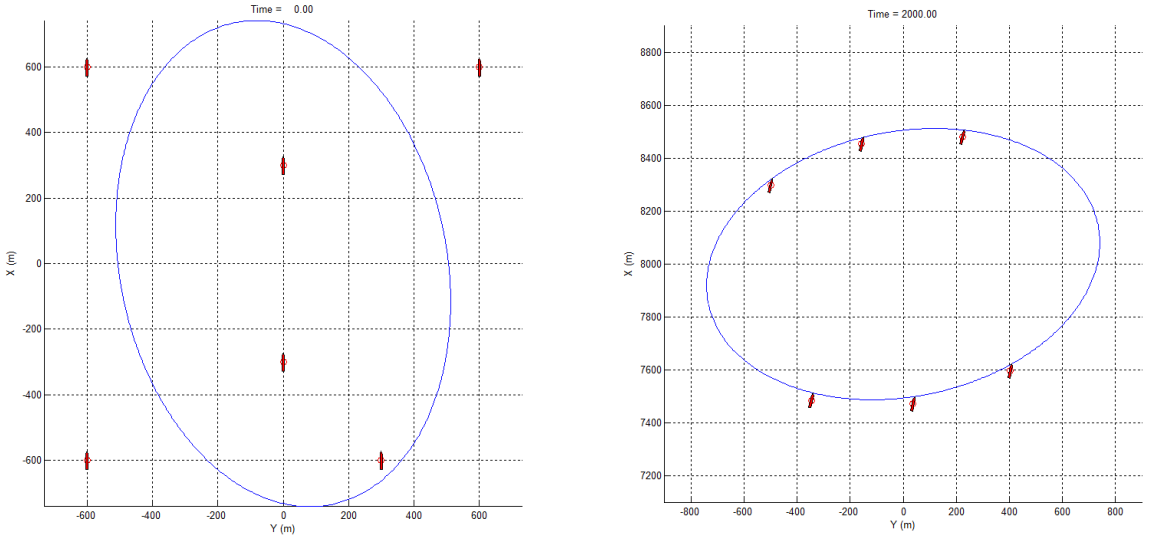
As our control group, we first tested an elliptical formation with equal values of a and b, effectively making a circle. Additionally, we had Curvature Optimization turned off for this test. As can be seen in Figure 47(b), the 20 m/s wind caused a large gap on the eastern side of the formation. The gap is not as pronounced in Figure 38(b) in simulation due to the increased self-avoidance controller gain, however there is still an increase in the steady-state swarm error. We will go into greater detail on the error later in this section.



(a) (b)

Figure 47 (a) Start and (b) End of Simulation with Circular Formation and no Curvature Optimization

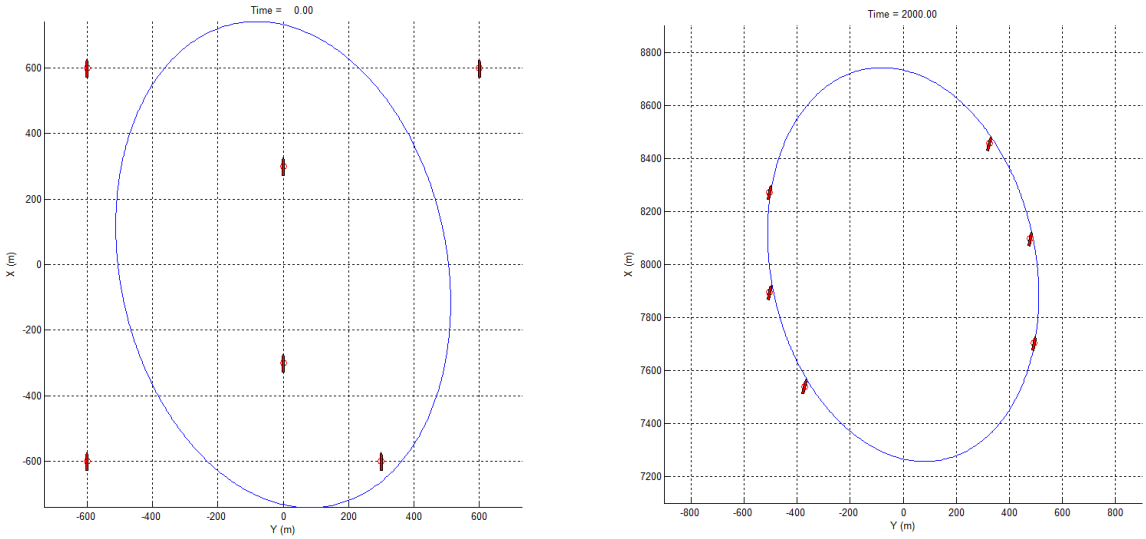
Next, we tested Method 1. As can be seen in Figure 48, the vessels started out in the same positions as in the previous test. By the end of the simulation, they have been able to get in formation and have moved into the regions commanded by the Curvature Optimization controller.



(a) (b)

Figure 48: (a) Start and (b) End of Simulation with Method 1 of Curvature Optimization and Complimentary Elliptical Formation

Last, we tested Method 2. As can be seen in Figure 49, the vessel's started out in the same positions as in the previous tests. By the end of the simulation, they have been able to get in formation and have moved into the regions commanded by the Curvature Optimization controller.



(a) (b)

Figure 49: (a) Start and (b) End of Simulation with Method 2 of Curvature Optimization and Complimentary Elliptical Formation

In Figure 50, we plotted the error in the three previous simulations versus time. Both Methods 1 and 2 improved the total steady-state error of the swarm. Using no disturbance correction, the total error was 13.3m. Using Methods 1 and 2, the total errors were was 5.19m and 10.8m, respectively.

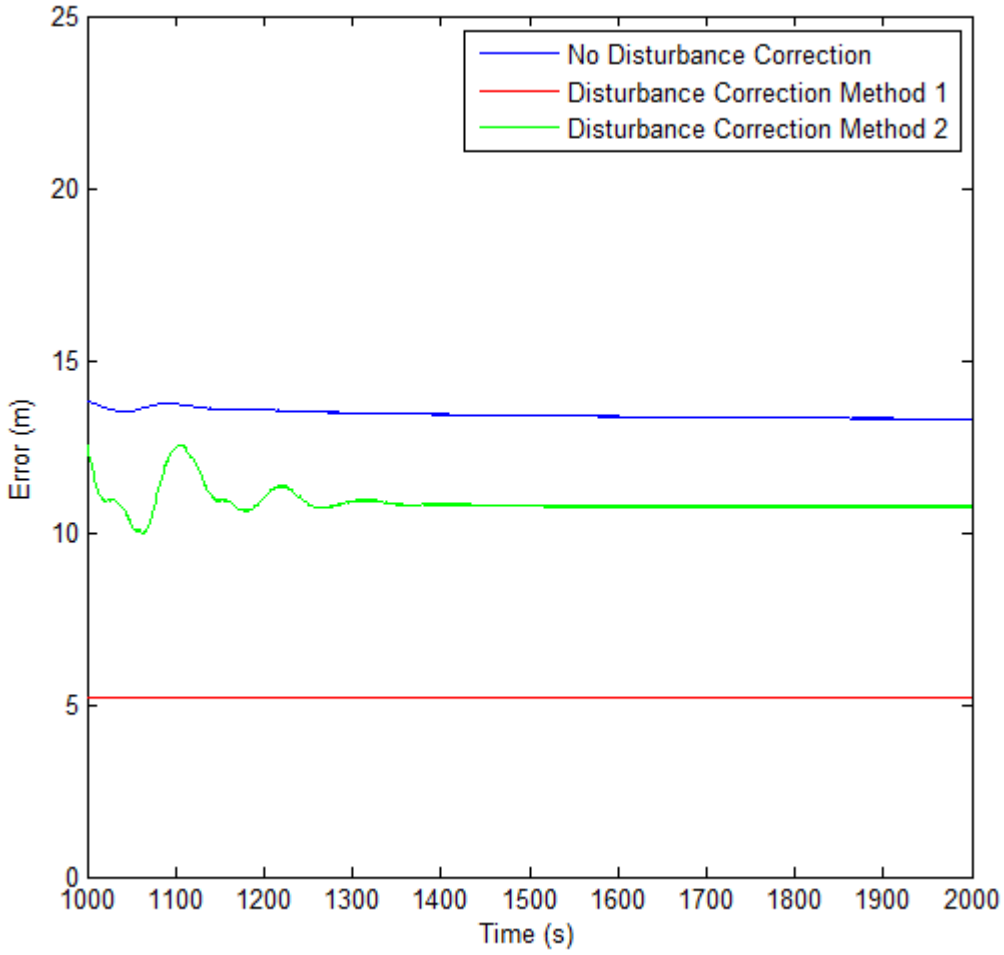


Figure 50: Total Error versus Time for Various Tests

Unlike in the simulations dealing with current, Method 1 appears to produce superior results. We found that this was due to the vessel's maneuvering dynamics. In situations where the formation's major axis was near-perpendicular to the direction of the environmental effect being studied, the vessel's rudder system had to be used to adjust for movement off of the formation. Using this model, the vessel reacted relatively slowly to rudder effects and the error in the model increased as a result. Such cases can be seen in Figure 44(b), where the major axis formed a minimum angle of 75° with the current heading in a direction of 270°, and in Figure 49(b), where the major axis formed a minimum angle of 79° with the wind coming from the direction of 090°.

In situations where the major axis of the formation was near-parallel to the environmental effect being studied, the vessel's propeller thrust system had to be used to account for movement off of the formation. Using this model, the vessel reacted more quickly to propeller thrust effects and the error in the model increased as a result. Such cases can be seen in Figure 45(b), where the major axis formed a minimum angle of 15°

with the current going in the direction of 270°, and in Figure 48(b), where the major axis formed a minimum angle of 11° with the wind coming from a direction of 090°.

To adequately correct for environmental disturbances, the swarm should select whichever method forms the smallest angle with the direction of the environmental effect.

IX. Conclusion

In this study, we developed a new vessel model containing more realistic maneuvering dynamics than models used in previous swarm studies. This model combines non-linear maneuvering dynamics with realistic environmental effects. To accomplish vessel-level control, we designed and implemented a controller which causes the vessel to achieve a desired velocity using a control-point method. This controller allows for both trajectory following as well as path following.

To accomplish, swarm-level control, we used a redundant manipulator formulation which allows the swarm to effectively coordinate multiple vessels and multiple tasks. This controller allowed the vessels to move in both circular and elliptical formations with relatively low levels of steady-state error.

After analyzing the effects of current on the swarm's ability to move in formation, we created a method which allows for the mitigation of these effects. The first part of this method involves a controller which moves the vessel to a location on the swarm where current has a less deleterious effect on formation following. This second part of this method allows for modification the swarm function by rotating an elliptical formation based on the velocities of the swarm and the current. This method allows the vessels to maintain dispersion on the formation and also decreases the steady-state swarm error. It is the intent of this study that the techniques developed in this paper will later be integrated into the design and building of a physical swarm by the Atlantic Center for the Innovative Design and Control of Small Ships (ACCeSS).

References

[1] Deptartment of the Navy, "The Navy Unmanned Surface Vehicle (USV) Master Plan," 2007.

[2] T. I. Fossen, Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs, and Underwater Vehicles, Trondheim, Norway: Marine Cybernetics, 2002.

[3] E. Borhaug, A. Pavlov and K. Y. Pettersen, "Cross-track formation control of underactuated surface vessels," *45th IEEE Conference on Decision and Control,* pp. 5955-5961, 2006.

[4] J. Ghommam and F. Mnif, "Coordinated Path-Following Control for a Group of Underactuated Surface Vessels," *IEEE Transactions on Industrial Electronics,* vol. 56, no. 10, pp. 3951-3963, 2009.

[5] H. Hooyer, Behavior and Handling of Ships, Centreville, Maryland: Cornell Maritime Press, 1983.

[6] M. R. Mainal and M. S. Kamil, "Estimation of ship manoeuvring characteristics in the conceptual design stage," *Jurnal Mekanikal,* pp. 44-60, 1996.

[7] E. Borhaug, A. Pavlov and K. Y. Pettersen, "Straight line path following for formations of underactuated underwater vehicles," *46th IEEE Conference on Decision and Control,* pp. 2905-2912, 2007.

[8] "Marine Systems Simulator," Marine Control, 2010. [Online]. Available: http://www.marinecontrol.org. [Accessed 25 September 2011].

[9] Y. C. Tan and B. E. Bishop, "Combining Classical and Behavior-Based Control for Swarms of Cooperating Vehicles," *2005 IEEE International Conference on Robotics and Automation,* pp. 2499-2504, 2005.

[10] R. M. Isherwood, "Wind Resistance of Merchant Ships," *RINA Trans,* vol. 15, pp. 327-338, 1972.

[11] N. W. J. Benstead, *An Investigation Into Ship Stabilisation Through Wave Prediction,* London: Marine Eng., Univ. College London, 2004.

[12] T. I. Fossen and T. Perez, "Modelling and Simulation of Environmental Disturbances," 18 Oct 2007. [Online]. Available: http://www.marinecontrol.org/pdf/Tutorial/CAMS_M5_disturbances.pdf. [Accessed 9 April

2012].

[13] B. E. Bishop, "Formation Control of Underactuated Autonomous Surface Vessels Using Redundant Manipulator Analogs," to appear in *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Paul, Minnesota, 2012.

[14] R. C. Arkin, Behavior-Based Robotics, Cambridge, MA: MIT Press, 1998.

Appendix A: Isherwood Tables

γ_r(deg)	A_0	A_1	A_2	A_3	A_4	A_5	A_6
0	2.152	-5.00	0.243	-0.164	0	0	0
10	1.714	-3.33	0.145	-0.121	0	0	0
20	1.818	-3.97	0.211	-0.143	0	0	0.033
30	1.965	-4.81	0.243	-0.154	0	0	0.041
40	2.333	-5.99	0.247	-0.190	0	0	0.042
50	1.726	-6.54	0.189	-0.173	0.348	0	0.048
60	0.913	-4.68	0	-0.104	0.482	0	0.052
70	0.457	-2.88	0	-0.068	0.346	0	0.043
80	0.341	-0.91	0	-0.031	0	0	0.032
90	0.355	0	0	0	-0.247	0	0.018
100	0.601	0	0	0	-0.372	0	-0.020
110	0.651	1.29	0	0	-0.582	0	-0.031
120	0.564	2.54	0	0	-0.748	0	-0.024
130	-0.142	3.58	0	0.047	-0.700	0	-0.028
140	-0.677	3.64	0	0.069	-0.529	0	-0.032
150	-0.723	3.14	0	0.064	-0.475	0	-0.032
160	-2.148	2.56	0	0.081	0	1.27	-0.027
170	-2.707	3.97	-0.175	0.126	0	1.81	0
180	-2.529	3.76	-0.174	0.128	0	1.55	0

γ_r(deg)	B_0	B_1	B_2	B_3	B_4	B_5	B_6
0	0	0	0	0	0	0	0
10	0.096	0.22	0	0	0	0	0
20	0.176	0.71	0	0	0	0	0
30	0.225	1.38	0	0.023	0	-0.29	0
40	0.329	1.82	0	0.043	0	-0.59	0
50	1.164	1.26	0.121	0	-0.242	-0.95	0
60	1.163	0.96	0.101	0	-0.177	-0.88	0
70	0.916	0.53	0.069	0	0	-0.65	0
80	0.844	0.55	0.082	0	0	-0.54	0
90	0.889	0	0.138	0	0	-0.66	0
100	0.799	0	0.155	0	0	-0.55	0
110	0.797	0	0.151	0	0	-0.55	0
120	0.996	0	0.184	0	-0.212	-0.66	0.34
130	1.014	0	0.191	0	-0.280	-0.69	0.44
140	0.784	0	0.166	0	-0.209	-0.53	0.38
150	0.536	0	0.176	-0.029	-0.163	0	0.27
160	0.251	0	0.106	-0.022	0	0	0
170	0.125	0	0.046	-0.012	0	0	0

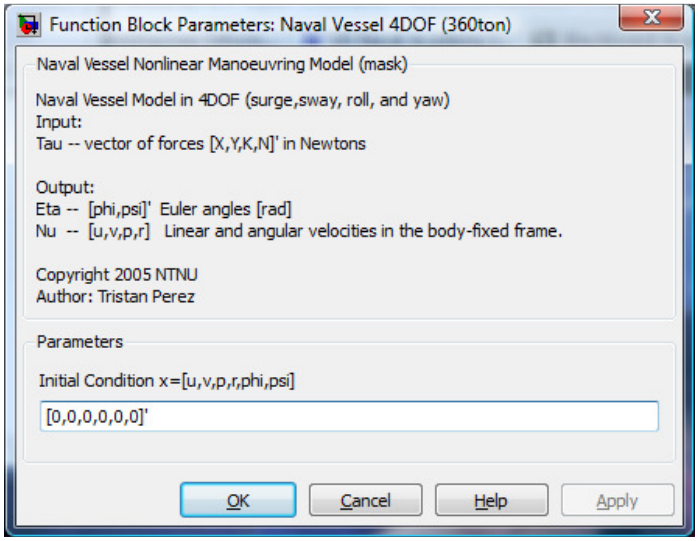
180	0	0	0	0	0	0	0

γ_r(deg)	C_0	C_1	C_2	C_3	C_4	C_5
0	0	0	0	0	0	0
10	0.0596	0.061	0	0	0	-0.074
20	0.1106	0.204	0	0	0	-0.170
30	0.2258	0.245	0	0	0	-0.380
40	0.2017	0.457	0	0.0067	0	-0.472
50	0.1759	0.573	0	0.0118	0	-0.523
60	0.1925	0.480	0	0.0115	0	-0.546
70	0.2133	0.315	0	0.0081	0	-0.526
80	0.1827	0.254	0	0.0053	0	-0.443
90	0.2627	0	0	0	0	-0.508
100	0.2102	0	-0.0195	0	0.0335	-0.492
110	0.1567	0	-0.0258	0	0.0497	-0.457
120	0.0801	0	-0.0311	0	0.0740	-0.396
130	-0.0189	0	-0.0488	0.0101	0.1128	-0.420
140	0.0256	0	-0.0422	0.0100	0.0889	-0.463
150	0.0552	0	-0.0381	0.0109	0.0689	-0.476
160	0.0881	0	-0.0306	0.0091	0.0366	-0.415
170	0.0851	0	-0.0122	0.0025	0	-0.220
180	0	0	0	0	0	0

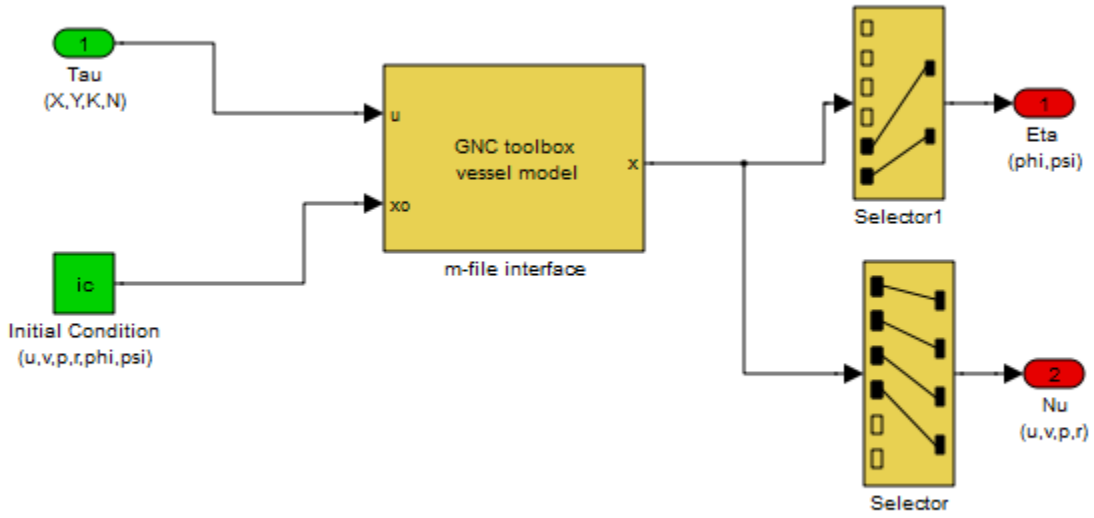
Appendix B: Vessel Dynamics Model

Vessel Model

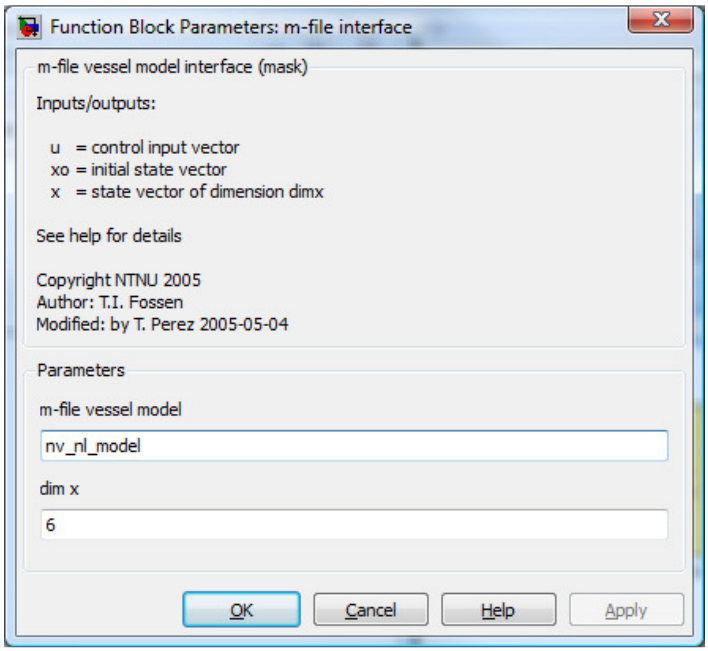
Simulink "Mask" For Vessel Model



Vessel Model Sybsystem



M-File Interface



Vessel Dynamics Code

```
function [out] = nv_nlin_model(in)
%Noninear Model for describing  surge, sway, roll and yaw Interactions
%of a multipurpose naval vessel. The surge is only coupled through a
%centripetal terms.
%
%Use:[out]=nv_lin_model(in)
%
%Output:
```

```
% out= M^-1 [X,Y,K,N,p,r]'
%
%where
% M is the total mass matrix
% X is the surge force (hydrodynamic+centriperal+external)
% Y is the sway force (hydrodynamic+centriperal+external)
% K is the sway force (hydrodynamic+centriperal+external)
% N is the sway force (hydrodynamic+centriperal+external)
% p is the roll rate
% r is the yaw rate
%
%Inputs:
% in=[u,v p r phi psi,Xe,Ye,Ke,Ne]'
%
%where
% u     = surge velocity           (m/s)
% v     = sway velocity           (m/s)
% p     = roll velocity           (rad/s)
% r     = yaw velocity            (rad/s)
% phi   = roll angle              (rad)
% psi   = yaw angle               (rad)
%
% U     = surge speed of the vessel [m/sec].
% Xe is the surge external force (eg rudder and fin force)
% Ye is the sway external force
% Ke is the sway external force
% Ne is the sway external force

% Reference: Blanke M. and Christensen A. (1993) "Rudder-roll
% damping autopilot robustness to sway-yaw-roll couplings."
% 10th Ship Control Systems Symposium, Ottawa, Canada.
%
%
% Notes: 1 - The model does not include rudder Machinery.
%        2 - The parameters of the model should be defined
%            in the structures h and const before using
%            the function.
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Copyright (C)  A. G. Jensen and M.S.Chislett (C) 1983-89,
%               Mogens Blanke and Antonio Tiano (C) 1996,
%               Mogens Blanke (C) 1997, 2001, 2003,
%
%
%(#)  Department  fo Automation
%       Danish Technical University, DTU.
%       DK 2800  Kgs. Lyngby, Denmark
%
%       Email: blanke@iau.dtu.dk
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Created:
%   Original models from A. G. Jensen and M.S.Chislett
%   (Danish Maritime Institute) 1983-89
%     Adapted for Matlab by Mogens Blanke and Antonio Tiano 1996
%   Modified for Matlab 5.3 implementation by Mogens Blanke 1997
%   Modified for Simulink by Mogens Blanke# and Tristan Perez*, 2001
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THIS VERSION modified by: Tristan Perez
% (*)  Centre For Ships and Ocean Structures
%      The  Norwegian University of Science and Technology NTNU
%       Univerisity dr. Callaghan, 2308 NSW, Austarlia,
%
% Email: tristan.perez@ntnu.no
% Date: 2005-05-04
% Comments:
%Adapted from the files reference (*) to match the data of the vessel
%design of ADI-Limited Australia.
%
% (*) Blanke M. and Christensen A. (1993)
%"Rudder-roll damping autopilot
% robustness to sway-yaw-roll couplings."
% 10th Ship Control Systems Symposium,
% Ottawa, Canada.
%
% _____________________________________________________________________
%
% MSS GNC is a Matlab toolbox for guidance, navigation and control.
% The toolbox is part of the Marine Systems Simulator (MSS).
%
% Copyright (C) 2008 Thor I. Fossen and Tristan Perez
%
% This program is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope that it will be useful, but
% WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with this program.  If not, see <http://www.gnu.org/licenses/>.
%
% E-mail: contact@marinecontrol.org
% URL:    <http://www.marinecontrol.org>

%Vessel Data
% struct const.
const.rho_water     =   1014.0;             %   water density     [kg/m^3]
const.rho_air           =     1.225 ;         % air density       [kg/m^3]
const.g                 =     9.81;         %   gravity constant
      [m/s^2]
const.deg2rad           =     pi/180;             %   degrees to radians
```

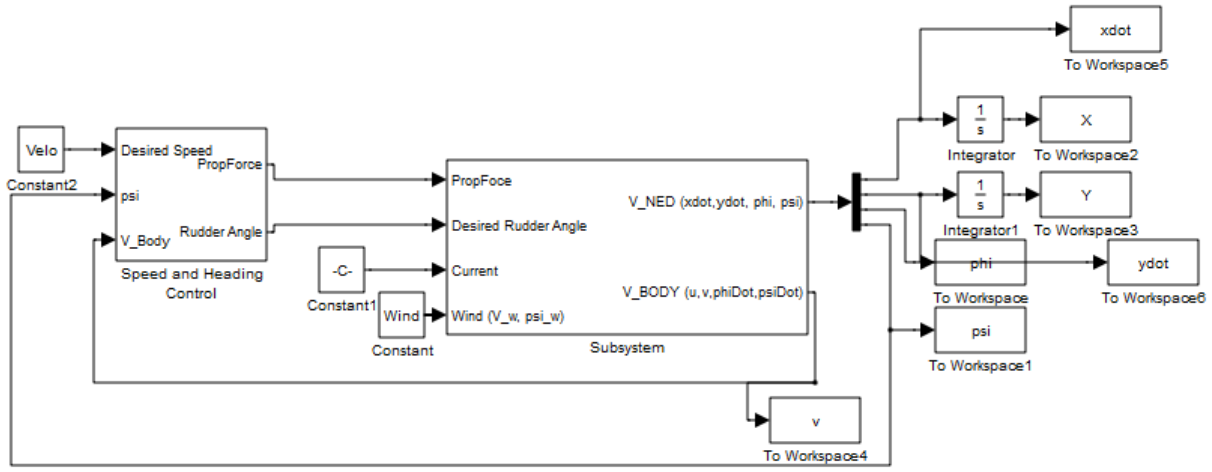
```
const.rad2deg           =     180/pi;             %   rad to degrees
const.ms2kt             =     3600/1852;      %       m/s to kt
const.kt2ms             =     1852/3600;      % kt to m/s
const.RPM2rads          =     2*pi/60;        % RPM to rad/s
const.rads2RPM          =   60/[2*pi];        % rad/s to RPM
const.HP2W              =     745.700;        % HP to Watt

%
%Struct rudder  (Modified by T.Perez)
rudder.sp    =1.5;                    %span
rudder.A     =1.5;                    %Area
rudder.ar    =3;                      %aspect ratio
rudder.dCL   =0.054; % 1/deg          %dCL/d a_e
rudder.stall =23;                     %a_stall

% Main Particulars (Modified by T.Perez)
h.Lpp    =  51.5 ;                   %Length between perpendiculars [m]
h.B      =  8.6  ;                   %Beam over all  [m]
h.D        =  2.3  ;                   %Draught [m]

%Load condition (Modified by T.Perez)
h.disp   =  357.0;                    %Displacement  [m^3]
h.m      =  h.disp*const.rho_water;   %Mass [Kg]
h.Izz    =  47.934*10^6 ;             %Yaw Inertia
h.Ixx    =  2.3763*10^6 ;             %Roll Inertia
h.U_nom  =  8.0   ;                       %Speed nominal [m/sec] (app 15kts)
h.KM        =  4.47;                  %  [m] Transverse metacentre above keel
h.KB        =  1.53;                  %  [m] Transverse centre of bouancy
h.gm        =  1.1;                       %  [m] Transverse Metacenter
h.bm        =  h.KM - h.KB;
h.LCG       = 20.41 ;                 % [m]      Longitudinal CG (from AP
considered at the rudder stock)
h.VCG       = 3.36  ;                 % [m]      Vertical  CG  above baseline
h.xG        = -3.38  ;                % coordinate of CG from the body fixed
frame adopted for the PMM test
h.zG            = -(h.VCG-h.D);           % coordinate of CG from the body
fixed frame adopted for the PMM test
h.m_xg          = h.m * h.xG;
h.m_zg          = h.m * h.zG;
h.Dp        = 1.6 ;                    % Propeller diameter [m]


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The hydrodynamic derivatives are given in dimensional form, and follow
% from the original publication of Blanke and Christensen 1993.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Data for surge equation
h.Xudot     = -17400.0 ;
h.Xuau      = -1.96e+003 ;
h.Xvr       =  0.33 * h.m ;

% Hydrodynamic coefficients in sway equation
h.Yvdot = -393000 ;
h.Ypdot = -296000 ;
```

```
h.Yrdot = -1400000 ;
h.Yauv  = -11800 ;
h.Yur   =  131000 ;
h.Yvav  = -3700 ;
h.Yrar   =  0 ;
h.Yvar  = -794000 ;
h.Yrav  = -182000 ;
h.Ybauv =  10800 ; % Y_{\phi |v u|}
h.Ybaur =  251000 ;
h.Ybuu  = -74 ;
%alpha rudder in degrees in the linear model (Modified by T.Perez)
h.Yduu  =  180*2*(.5*const.rho_water*rudder.A*rudder.dCL)/pi;

% Hydrodynamic coefficients in roll equation
h.Kvdot =  296000 ;
h.Kpdot = -774000 ;
h.Krdot =  0 ;
h.Kauv  =  9260 ;
h.Kur   = -102000 ;
h.Kvav  =  29300 ;
h.Krar  =  0 ;
h.Kvar  =  621000 ;
h.Krav  =  142000 ;
h.Kbauv =  -8400 ;
h.Kbaur =  -196000 ;
h.Kbuu  =  -1180 ;
h.Kaup  =  -15500 ;
h.Kpap  =  -416000 ;
h.Kp    =  -500000 ;
h.Kb    =  0.776*h.m*const.g;
h.Kbbb  =  -0.325*h.m*const.g ;

% Hydrodynamic coefficients in yaw equation*)
h.Nvdot =  538000 ;
h.Npdot =  0 ;
h.Nrdot = -38.7e6;
h.Nauv  = -92000 ;
h.Naur  = -4710000 ;
h.Nvav  =  0 ;
h.Nrar  = -202000000 ;
h.Nvar  =  0 ;
h.Nrav  = -15600000 ;
h.Nbauv = -214000 ;
h.Nbuar = -4980000 ;
h.Nbuau = -8000 ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Rename inputs of the function
u   = in(1);
v   = in(2);
p     = in(3);
r     = in(4);
b     = in(5);% b -denoted phi roll
psi = in(6);
```

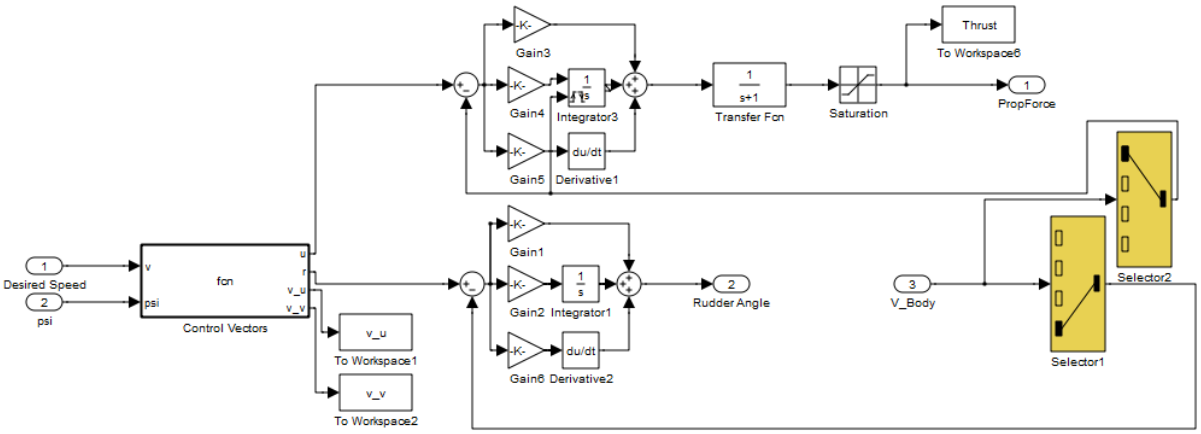
```
Xe   = in(7);
Ye  = in(8); %External forces
Ke  = in(9);
Ne  = in(10);
%Auxiliary variables
au = abs(u);
av = abs(v);
ar = abs(r);
ap = abs(p);
ab = abs(b);
L2 = h.Lpp^2;

%Total Mass Matrix
M=[ (h.m-h.Xudot)  0   0   0   0   0;
   0 (h.m-h.Yvdot) -(h.m*h.zG+h.Ypdot) (h.m*h.xG-h.Yrdot) 0 0;
   0 -(h.m*h.zG+h.Kvdot) (h.Ixx-h.Kpdot) -h.Krdot 0 0;
   0 (h.m*h.xG-h.Nvdot) -h.Npdot (h.Izz-h.Nrdot) 0 0;
   0 0 0 0 1 0;
   0 0 0 0 0 1] ;
%Hydrodynamic forces without added mass terms (considered in the M matrix)
Xh  = h.Xuau*u*au+h.Xvr*v*r;

Yh = h.Yauv*au*v + h.Yur*u*r + h.Yvav*v*av + h.Yvar*v*ar + h.Yrav*r*av ...
   + h.Ybauv*b*abs(u*v) + h.Ybaur*b*abs(u*r) + h.Ybuu*b*u^2;

Kh = h.Kauv*au*v +h.Kur*u*r + h.Kvav*v*av + h.Kvar*v*ar + h.Krav*r*av ...
   + h.Kbauv*b*abs(u*v) + h.Kbaur*b*abs(u*r) + h.Kbuu*b*u^2 + h.Kaup*au*p...
   + h.Kpap*p*ap +h.Kp*p +h.Kbbb*b^3-(const.rho_water*const.g*h.gm*h.disp)*b;

Nh = h.Nauv*au*v + h.Naur*au*r + h.Nrar*r*ar + h.Nrav*r*av...
   +h.Nbauv*b*abs(u*b) + h.Nbuar*b*u*ar + h.Nbuau*b*u*au;


%Rigid-body centripetal accelerations
Xc =   h.m*(r*v+h.xG*r^2-h.zG*p*r);
Yc = - h.m*u*r;
Kc =   h.m*h.zG*u*r;
Nc = - h.m*h.xG*u*r;

%Total forces
F1=Xh+Xc+Xe;
F2=Yh+Yc+Ye;
F4=Kh+Kc+Ke;
F6=Nh+Nc+Ne;

out=M\[F1; F2; F4; F6; p; r];
%EOF
```

Appendix C: Course Following Simulation

Course Following Simulink Model



Speed and Heading Control



Desired Course Control Code

```
% CourseFollowingControl.m
% Controls Course Following simulation

% Thrust Control Gains
Ks_p = 5000;
Ks_i = 200;
Ks_d = 0;

% Rudder Control Gains
```

```
Kh_p = -15;
Kh_i = 0;
Kh_d = 0;

% Desired Velocity
Velo = [.707*4,.707*4];

% Environmental
Wind = [0,0];
Current = [0,0];

% Tracking Point Control Gain
k_d = .1;

% Stop Time of Simulation
tStop = 1000;

sim('CourseFollowingModel')

CourseFollowingPlotter(X,Y,psi,tout,rudderAngle,Velo,v_u,v_v)
```

Plotting code for Course Control Simulation

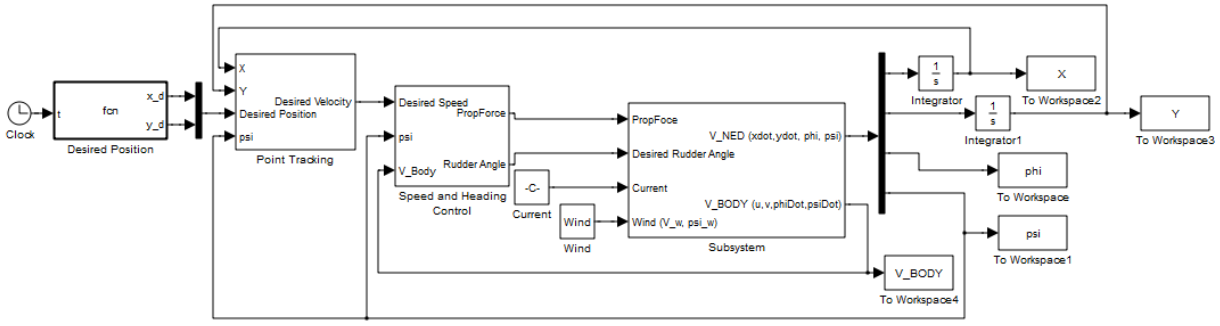
```
function CourseFollowingPlotter(X,Y,psi,tout,rudderAngle,x_d,y_d)
% X -  n by 1 array with X positions(meters) of vessel thorughout sim
% Y -  n by 1 array with Y positions(meters) of vessel thorughout sim
% psi -  n by 1 array with yaw angles(rad) of vessel thorughout sim
% tout -  n by 1 array with Simulink time output values
% rudderAngle -  n by 1 array with rudder angles
% x_d - n by 1 matrix
% y_d - n by 1 matrix

fig1 = figure(1);
clf
hold on

% Plots Ship Image
ship_x=X(1);
ship_y=Y(1);
yaw=psi(1);
x_points = [8.6/2 -8.6/2 -8.6/2 0 8.6/2 8.6/2];
y_points = [-55/2 -55/2 2/3*55/2 55/2 2/3*55/2 -55/2];
ship_points = [y_points;
    x_points;
    zeros(1,6)];

rudder_x = [0 5*sin(0)];
rudder_y = [-55/2 -55/2-5*cos(0)];
rudder_points = [rudder_y;
    rudder_x;
    zeros(1,2)];

R = [cos(yaw) -sin(yaw) 0;
    sin(yaw)  cos(yaw) 0;
    0         0        1];
```

```
ship_view = R*ship_points;
rudder_view = R*rudder_points;

shipImage =  fill(ship_view(2,:)+ship_y,ship_view(1,:)+ship_x,'r');
rudderImage = plot(rudder_view(2,:)+ship_y,rudder_view(1,:)+ship_x);
shipLocation = plot(ship_y,ship_x,'o');

% Plot's Ship Path
shipPath = plot(Y(1),X(1),'r--');

% Plot's Tracking Point and Desired Path
trackPoint = plot(y_d(1),x_d(1),'*');
plot(y_d,x_d)

axis equal;

XLabel('Y (meters)');
YLabel('X (meters)');

for(i=1:100:length(tout))
    % Updates Ship Image
    ship_x=X(i);
    ship_y=Y(i);
    yaw=psi(i);

    R = [cos(yaw) -sin(yaw) 0;
        sin(yaw)  cos(yaw) 0;
        0         0        1];

    rudder_x = [0 5*sin(-rudderAngle(i))];
    rudder_y = [-55/2 -55/2-5*cos(-rudderAngle(i))];
    rudder_points = [rudder_y;
        rudder_x;
        zeros(1,2)];

    ship_view = R*ship_points;
    rudder_view = R*rudder_points;

    set(shipImage,'XData',ship_view(2,:)+ship_y);
    set(shipImage,'YData',ship_view(1,:)+ship_x);
    set(rudderImage,'XData',rudder_view(2,:)+ship_y);
    set(rudderImage,'YData',rudder_view(1,:)+ship_x);
    set(shipLocation,'XData',Y(i));
    set(shipLocation,'YData',X(i));

    % Update Ship's Path
    set(shipPath,'XData',Y(1:i));
    set(shipPath,'YData',X(1:i));

    % Updates Tracking Point Position
    set(trackPoint,'XData',y_d(i));
    set(trackPoint,'YData',x_d(i));
```

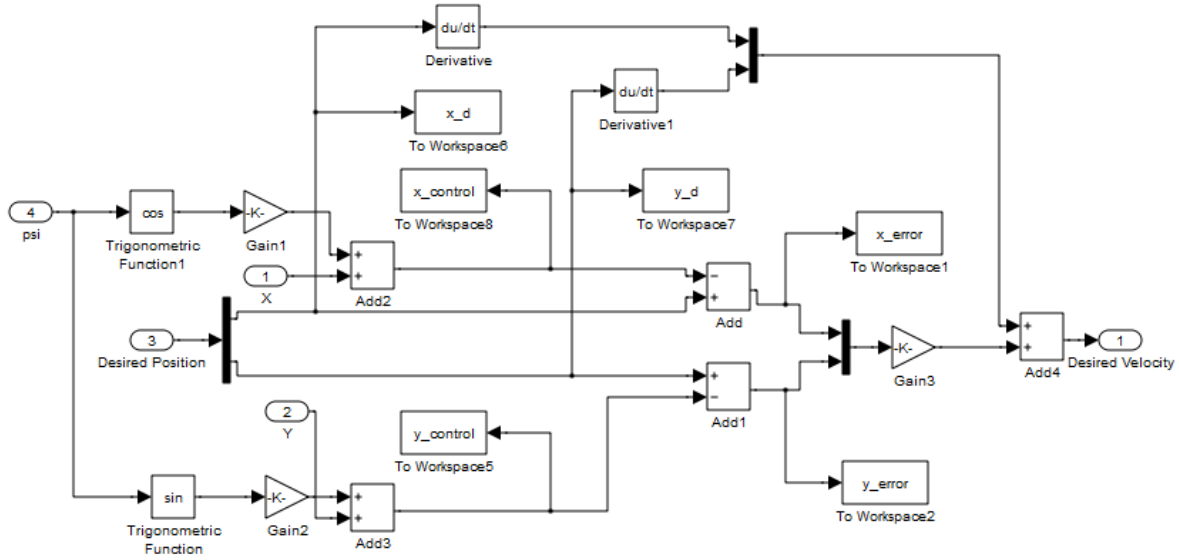
```
    % Updates Time Stamp in Simulation
    FigTitle =sprintf('Time = %7.2f',tout(i));
    Title(FigTitle);

    pause()
end
```

Appendix D: Trajectory Tracking Simulation

Trajectory Tracking Simulink Model



Point Tracking Control



Desired Position Code

```
function [x_d,y_d]  = fcn(t)
%% Creates Tracking Point which Moves in a Circular Path
w = 1/900*2*pi;

x_d = 500*cos(w*t);
y_d = 500*sin(w*t);
```

Control Code for Trajectory Simulation

```
% TrajectoryTrackingControl.m
% Plots results of TrajectroyTracking simulation

% Thrust Control Gains
```

```
Ks_p = 1000;
Ks_i = 5;
Ks_d = 0;

% Rudder Control Gains
Kh_p = -15;
Kh_i = 0;
Kh_d = 0;

% Environmental
Wind = [0,0];
Current = [0,0];

% Tracking Point Control Gain
k_d = .1;

% Stop Time of Simulation
tStop = 2500;

sim('TrajectoryTracking')

TrajectoryTrackingPlotter(X,Y,psi,tout,rudderAngle,x_d,y_d);
```

Plotting Code for Trajectory Simulation

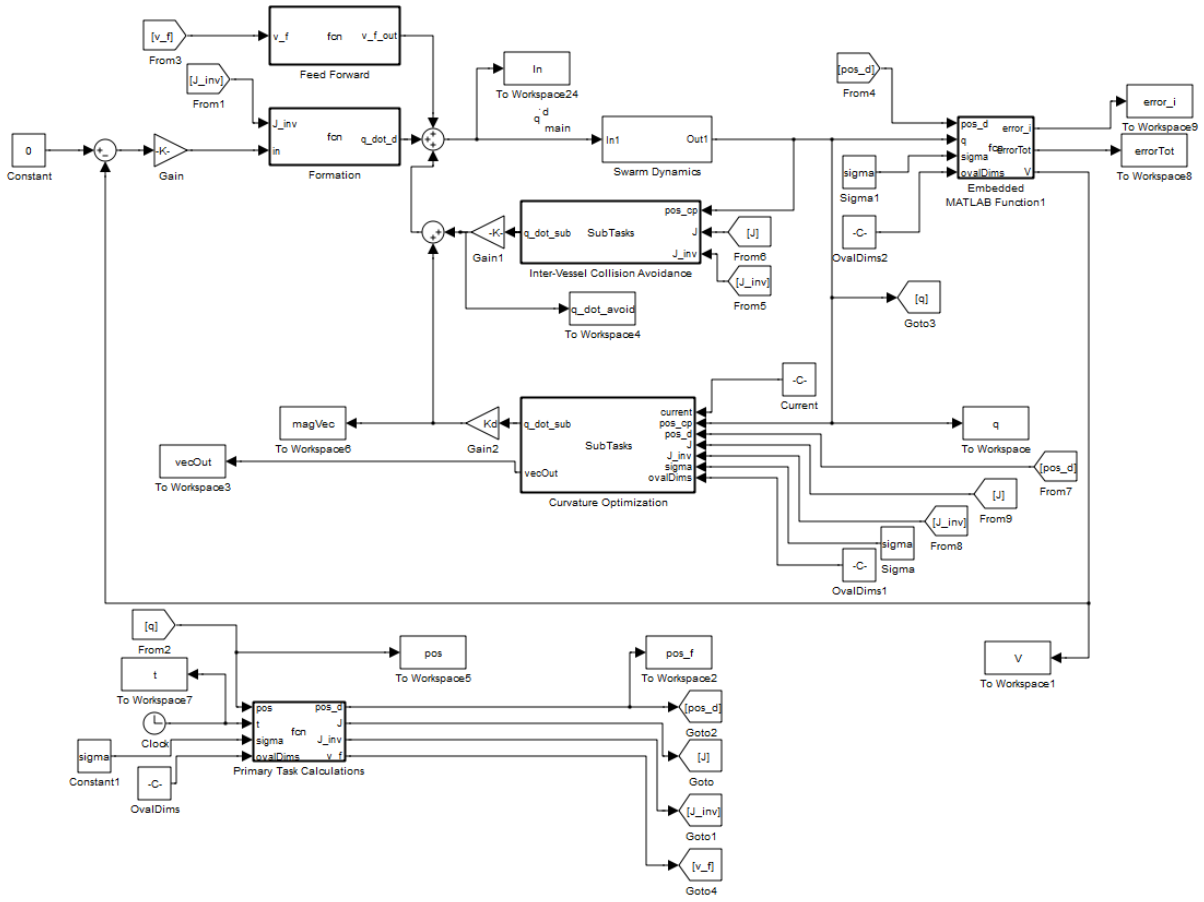
```
function TrajectoryTrackingPlotter(X,Y,psi,tout,rudderAngle,x_d,y_d)
% X -  n by 1 array with X positions(meters) of vessel thorughout sim
% Y -  n by 1 array with Y positions(meters) of vessel thorughout sim
% psi -  n by 1 array with yaw angles(rad) of vessel thorughout sim
% tout -  n by 1 array with Simulink time output values
% rudderAngle -  n by 1 array with rudder angles
% x_d - n by 1 matrix
% y_d - n by 1 matrix

fig1 = figure(1);
clf
hold on

% Plots Ship Image
ship_x=X(1);
ship_y=Y(1);
yaw=psi(1);
x_points = [8.6/2 -8.6/2 -8.6/2 0 8.6/2 8.6/2];
y_points = [-55/2 -55/2 2/3*55/2 55/2 2/3*55/2 -55/2];
ship_points = [y_points;
    x_points;
    zeros(1,6)];

rudder_x = [0 5*sin(0)];
rudder_y = [-55/2 -55/2-5*cos(0)];
rudder_points = [rudder_y;
    rudder_x;
    zeros(1,2)];
```

```
R = [cos(yaw) -sin(yaw) 0;
    sin(yaw)  cos(yaw) 0;
    0         0        1];

ship_view = R*ship_points;
rudder_view = R*rudder_points;

shipImage =  fill(ship_view(2,:)+ship_y,ship_view(1,:)+ship_x,'r');
rudderImage = plot(rudder_view(2,:)+ship_y,rudder_view(1,:)+ship_x);
shipLocation = plot(ship_y,ship_x,'o');

% Plot's Ship Path
shipPath = plot(Y(1),X(1),'r--');

% Plot's Tracking Point and Desired Path
trackPoint = plot(y_d(1),x_d(1),'*');
plot(y_d,x_d)

axis equal;

XLabel('Y (meters)');
YLabel('X (meters)');

for(i=1:100:length(tout))
    % Updates Ship Image
    ship_x=X(i);
    ship_y=Y(i);
    yaw=psi(i);

    R = [cos(yaw) -sin(yaw) 0;
        sin(yaw)  cos(yaw) 0;
        0         0        1];

    rudder_x = [0 5*sin(-rudderAngle(i))];
    rudder_y = [-55/2 -55/2-5*cos(-rudderAngle(i))];
    rudder_points = [rudder_y;
        rudder_x;
        zeros(1,2)];

    ship_view = R*ship_points;
    rudder_view = R*rudder_points;

    set(shipImage,'XData',ship_view(2,:)+ship_y);
    set(shipImage,'YData',ship_view(1,:)+ship_x);
    set(rudderImage,'XData',rudder_view(2,:)+ship_y);
    set(rudderImage,'YData',rudder_view(1,:)+ship_x);
    set(shipLocation,'XData',Y(i));
    set(shipLocation,'YData',X(i));

    % Update Ship's Path
    set(shipPath,'XData',Y(1:i));
    set(shipPath,'YData',X(1:i));

    % Updates Tracking Point Position
```

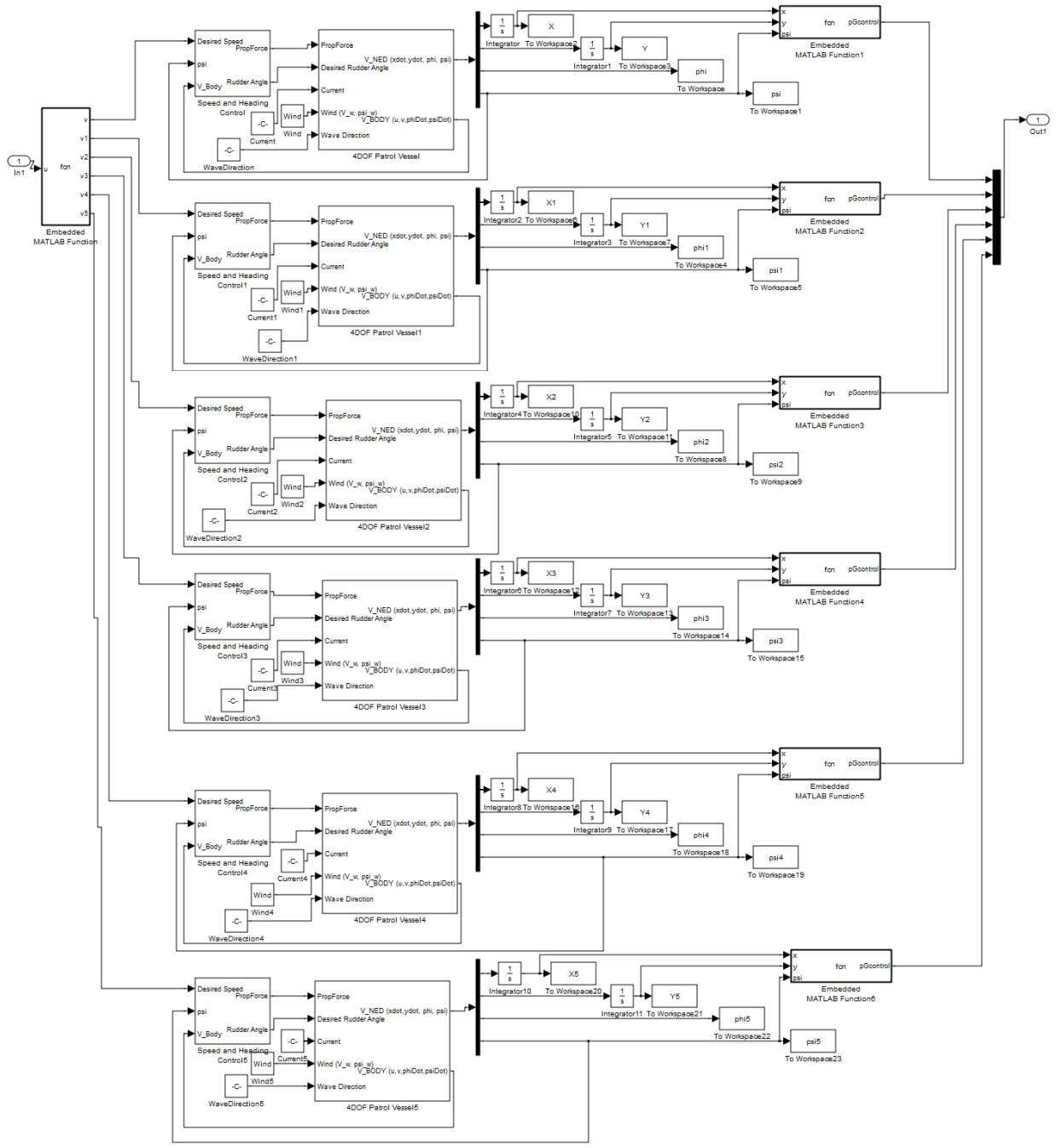
```
    set(trackPoint,'XData',y_d(i));
    set(trackPoint,'YData',x_d(i));

    % Updates Time Stamp in Simulation
    FigTitle =sprintf('Time = %7.2f',tout(i));
    Title(FigTitle);

    pause()
end
```

Appendix E: Final Swarm Simulation

Final Swarm Simulink Model



Swarm Dynamics Block



Primary Task Calculations

```
function [pos_d,J,J_inv,v_f] = fcn(pos,t,sigma,ovalDims)

% Oval Dimensions
a = ovalDims(1);
b = ovalDims(2);
```

```
% Formation Velocity
vx = 4;
vy = 0;
v_f = [vx;vy];

% Formation Position
pos_d = [vx*t,vy*t]';


% Vessel Coordinates
x1 = pos(1);
x2 = pos(3);
x3 = pos(5);
x4 = pos(7);
x5 = pos(9);
x6 = pos(11);

y1 = pos(2);
y2 = pos(4);
y3 = pos(6);
y4 = pos(8);
y5 = pos(10);
y6 = pos(12);


% Calculates Jacobian of Primary Task
J = [ 2*((2*cos(sigma)*(cos(sigma)*(x1 - t*vx) + sin(sigma)*(y1 - t*vy)))/a^2
      - (2*sin(sigma)*(cos(sigma)*(y1 - t*vy) - sin(sigma)*(x1 -
      t*vx)))/b^2)*((cos(sigma)*(x1 - t*vx) + sin(sigma)*(y1 - t*vy))^2/a^2 +
      (cos(sigma)*(y1 - t*vy) - sin(sigma)*(x1 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(y1 - t*vy) - sin(sigma)*(x1 - t*vx)))/b^2
      + (2*sin(sigma)*(cos(sigma)*(x1 - t*vx) + sin(sigma)*(y1 -
      t*vy)))/a^2)*((cos(sigma)*(x1 - t*vx) + sin(sigma)*(y1 - t*vy))^2/a^2 +
      (cos(sigma)*(y1 - t*vy) - sin(sigma)*(x1 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(x2 - t*vx) + sin(sigma)*(y2 - t*vy)))/a^2
      - (2*sin(sigma)*(cos(sigma)*(y2 - t*vy) - sin(sigma)*(x2 -
      t*vx)))/b^2)*((cos(sigma)*(x2 - t*vx) + sin(sigma)*(y2 - t*vy))^2/a^2 +
      (cos(sigma)*(y2 - t*vy) - sin(sigma)*(x2 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(y2 - t*vy) - sin(sigma)*(x2 - t*vx)))/b^2
      + (2*sin(sigma)*(cos(sigma)*(x2 - t*vx) + sin(sigma)*(y2 -
      t*vy)))/a^2)*((cos(sigma)*(x2 - t*vx) + sin(sigma)*(y2 - t*vy))^2/a^2 +
      (cos(sigma)*(y2 - t*vy) - sin(sigma)*(x2 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(x3 - t*vx) + sin(sigma)*(y3 - t*vy)))/a^2
      - (2*sin(sigma)*(cos(sigma)*(y3 - t*vy) - sin(sigma)*(x3 -
      t*vx)))/b^2)*((cos(sigma)*(x3 - t*vx) + sin(sigma)*(y3 - t*vy))^2/a^2 +
      (cos(sigma)*(y3 - t*vy) - sin(sigma)*(x3 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(y3 - t*vy) - sin(sigma)*(x3 - t*vx)))/b^2
      + (2*sin(sigma)*(cos(sigma)*(x3 - t*vx) + sin(sigma)*(y3 -
      t*vy)))/a^2)*((cos(sigma)*(x3 - t*vx) + sin(sigma)*(y3 - t*vy))^2/a^2 +
      (cos(sigma)*(y3 - t*vy) - sin(sigma)*(x3 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(x4 - t*vx) + sin(sigma)*(y4 - t*vy)))/a^2
      - (2*sin(sigma)*(cos(sigma)*(y4 - t*vy) - sin(sigma)*(x4 -
      t*vx)))/b^2)*((cos(sigma)*(x4 - t*vx) + sin(sigma)*(y4 - t*vy))^2/a^2 +
      (cos(sigma)*(y4 - t*vy) - sin(sigma)*(x4 - t*vx))^2/b^2 - 1),
```

```
      2*((2*cos(sigma)*(cos(sigma)*(y4 - t*vy) - sin(sigma)*(x4 - t*vx)))/b^2
      + (2*sin(sigma)*(cos(sigma)*(x4 - t*vx) + sin(sigma)*(y4 -
      t*vy)))/a^2)*((cos(sigma)*(x4 - t*vx) + sin(sigma)*(y4 - t*vy))^2/a^2 +
      (cos(sigma)*(y4 - t*vy) - sin(sigma)*(x4 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(x5 - t*vx) + sin(sigma)*(y5 - t*vy)))/a^2
      - (2*sin(sigma)*(cos(sigma)*(y5 - t*vy) - sin(sigma)*(x5 -
      t*vx)))/b^2)*((cos(sigma)*(x5 - t*vx) + sin(sigma)*(y5 - t*vy))^2/a^2 +
      (cos(sigma)*(y5 - t*vy) - sin(sigma)*(x5 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(y5 - t*vy) - sin(sigma)*(x5 - t*vx)))/b^2
      + (2*sin(sigma)*(cos(sigma)*(x5 - t*vx) + sin(sigma)*(y5 -
      t*vy)))/a^2)*((cos(sigma)*(x5 - t*vx) + sin(sigma)*(y5 - t*vy))^2/a^2 +
      (cos(sigma)*(y5 - t*vy) - sin(sigma)*(x5 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(x6 - t*vx) + sin(sigma)*(y6 - t*vy)))/a^2
      - (2*sin(sigma)*(cos(sigma)*(y6 - t*vy) - sin(sigma)*(x6 -
      t*vx)))/b^2)*((cos(sigma)*(x6 - t*vx) + sin(sigma)*(y6 - t*vy))^2/a^2 +
      (cos(sigma)*(y6 - t*vy) - sin(sigma)*(x6 - t*vx))^2/b^2 - 1),
      2*((2*cos(sigma)*(cos(sigma)*(y6 - t*vy) - sin(sigma)*(x6 - t*vx)))/b^2
      + (2*sin(sigma)*(cos(sigma)*(x6 - t*vx) + sin(sigma)*(y6 -
      t*vy)))/a^2)*((cos(sigma)*(x6 - t*vx) + sin(sigma)*(y6 - t*vy))^2/a^2 +
      (cos(sigma)*(y6 - t*vy) - sin(sigma)*(x6 - t*vx))^2/b^2 - 1)];

% Calculates Pseudoinverse of Primary Task
J_inv = J.'*(J*J.')^-1;
```

Inter-Vessel Collision Avoidance Code

```
function q_dot_sub = SubTasks(pos_cp,J,J_inv)

Cont_min = 0;
Cont_max = 750;
v_avoid_x = [0 0 0 0 0 0]';
v_avoid_y = [0 0 0 0 0 0]';
v_out = [0 0 0 0 0 0 0 0 0 0 0 0]';
dist = zeros(6);

for(this=1:6)
    for(others=1:6)
        xdif = pos_cp(2*this-1)-pos_cp(2*others-1);
        ydif = pos_cp(2*this)-pos_cp(2*others);
        dist(this,others) = sqrt(ydif^2+xdif^2);
        if (dist(this,others)<Cont_max)&&(this~=others)
            psi_obs = atan2(ydif,xdif);
            M_avoid = (Cont_max-dist(this,others))/(Cont_max-Cont_min);
            v_avoid_x(others) = M_avoid*cos(psi_obs);
            v_avoid_y(others) = M_avoid*sin(psi_obs);
        else
            v_avoid_x(others) = 0;
            v_avoid_y(others) = 0;
        end
        v_out(2*this-1) = sum(v_avoid_x);
        v_out(2*this) = sum(v_avoid_y);

    end
end
```

```
q_dot_sub = (eye(12)-J_inv*J)*v_out;
```

Curvature Optimization Code

```
function [q_dot_sub,vecOut]  = SubTasks(current,pos_cp,pos_d,J,J_inv,
      sigma,ovalDims)

a = ovalDims(1);
b = ovalDims(2);

Vec = [0 0; 0 0; 0 0; 0 0; 0 0; 0 0];
grad = [0 0; 0 0; 0 0; 0 0; 0 0; 0 0;];
psiMags = [0 0; 0 0; 0 0; 0 0; 0 0; 0 0;];
psiGrad = [0 0 0 0 0 0];
psiDiff = [0 0 0 0 0 0]';
psiVec = [0 0 0 0 0 0]';
path = [0 0 0 0 0 0]';

psiDis = sigma;


for (i = 1:1:6)
    grad(i,:) = [(2*cos(sigma)*(cos(sigma)*(pos_cp(2*i-1) - pos_d(1)) +
      sin(sigma)*(pos_cp(2*i) - pos_d(2))))/a^2 -
      (2*sin(sigma)*(cos(sigma)*(pos_cp(2*i) - pos_d(2)) -
      sin(sigma)*(pos_cp(2*i-1) - pos_d(1))))/b^2;
                 (2*cos(sigma)*(cos(sigma)*(pos_cp(2*i) - pos_d(2)) -
      sin(sigma)*(pos_cp(2*i-1) - pos_d(1))))/b^2 +
      (2*sin(sigma)*(cos(sigma)*(pos_cp(2*i-1) - pos_d(1)) +
      sin(sigma)*(pos_cp(2*i) - pos_d(2))))/a^2]';
    psiGrad(i) = atan2(grad(i,2),grad(i,1));

    psiDiff(i) = psiGrad(:,i)-psiDis;

    if psiDiff(i)<-pi
        psiDiff(i)= psiDiff(i)+2*pi;
    end
    if psiDiff(i)>pi
        psiDiff(i)= psiDiff(i)-2*pi;
    end

    if (psiDiff(i)<0)
        if abs(psiDiff(i))<pi/2
            psiVec(i) = psiGrad(i)+pi/2;
            path(i) = 1;
        else
            psiVec(i) = psiGrad(i)-pi/2;
            path(i) = 2;
        end
    else
        if abs(psiDiff(i))<pi/2
            psiVec(i) = psiGrad(i)-pi/2;
            path(i) = 3;
        else
            psiVec(i) = psiGrad(i)+pi/2;
```

```
            path(i) = 4;
        end
    end

    if psiVec(i)<-pi
        psiVec(i)= psiVec(i)+2*pi;
    end
    if psiVec(i)>pi
        psiVec(i)= psiVec(i)-2*pi;
    end

    psiMags(i,:) = [psiDiff(i),psiDiff(i)+pi];
    if psiMags(i,2)>pi
        psiMags(i,2) = psiMags(i,2)-2*pi;
    end

    Vec(i,:)= sin(min(abs(psiMags(i,:))))*[cos(psiVec(i)),sin(psiVec(i))];
end

vecOut =[Vec(1,:)';Vec(2,:)';Vec(3,:)';Vec(4,:)';Vec(5,:)';Vec(6,:)'];

q_dot_sub = (eye(12)-
      J_inv*J)*[Vec(1,:)';Vec(2,:)';Vec(3,:)';Vec(4,:)';Vec(5,:)';Vec(6,:)'];
```